



PI SIMULATION
Build and test projects
without lifting a finger



FASTER DRIVES
Upgrade to a super-
fast multi-SSD array



AI MADE EASY
Make testing LLMs as
simple as any install

LINUX FORMAT

The **#1** open source mag



SYSTEM CONTROL!
Connect to your PCs from
anywhere in the world

FRESH MINT 22!

Update your PC to the slickest Ubuntu spin
● New apps ● Classic desktop ● Matrix chat
● Secure Flatpak ● The latest 6.8 kernel

100
pages of Linux
tricks, tips
& more!



PLUS: HOW TO

- Code a fun Wordle-style game
- Try the cutting-edge NixOS
- Run the classic Symbian OS from Psion

INSIDE PIPEWIRE

Control audio and video
the modern Linux way!

STEAM STREAMS

Recording games has
never been so easy

C64 ASSEMBLER

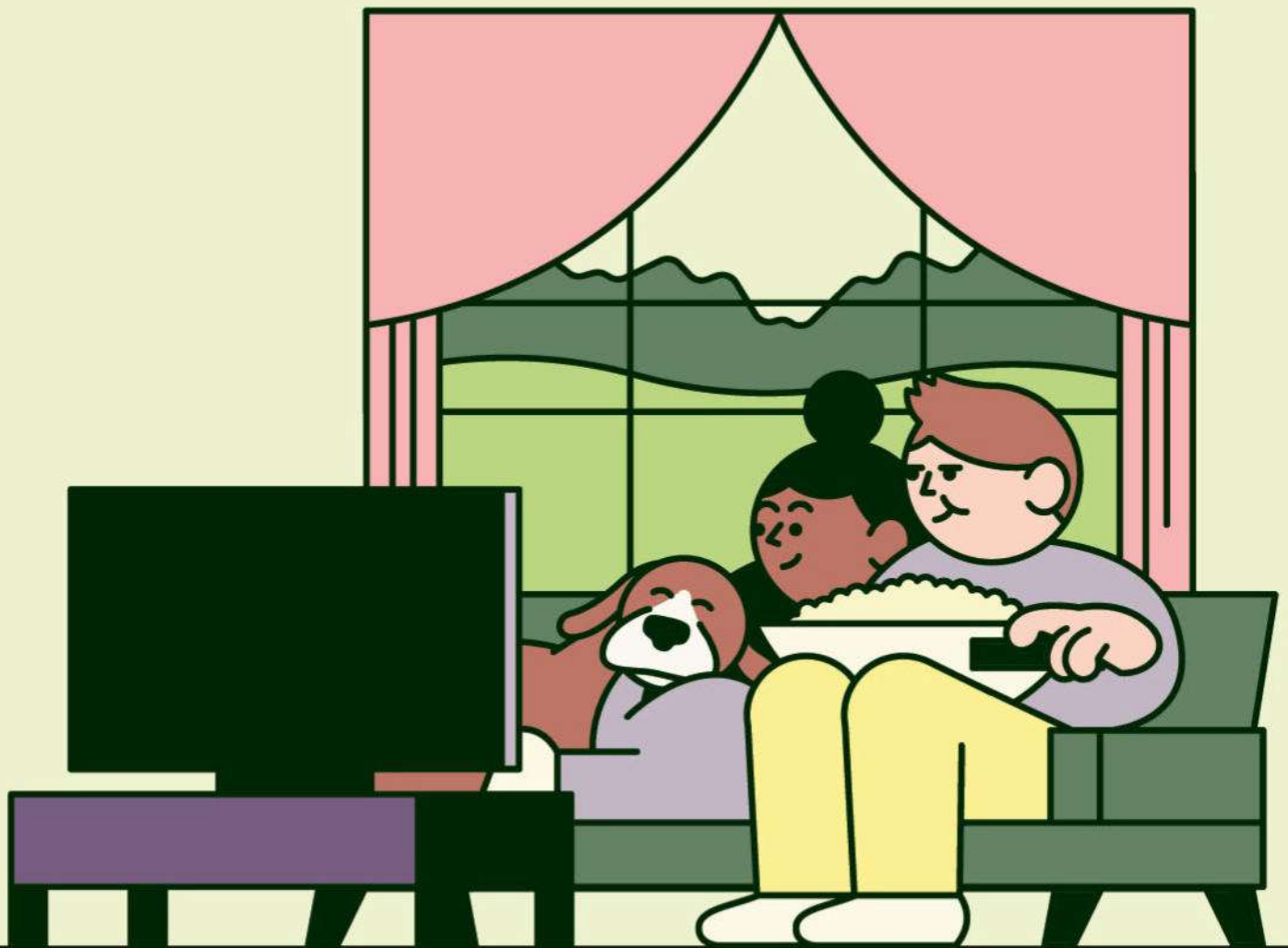
Learn how to compile
C64 machine code

LXF September 2024



SAVVY SAVERS, STEP THIS WAY.

Discover cover options for car, home and travel insurance, plus broadband and more.



GO.
COMPARE

Get more information and compare quotes at [Go.Compare](https://www.go.compare).



LINUX FORMAT



» MEET THE TEAM

This issue, we're introducing readers to Linux Mint 22 and its Cinnamon desktop; in the name of open source choice, what other desktop should people try and why?



Jonni Bidwell

I'm agog about the upcoming alpha of Pop!_ OS's Cosmic desktop. But there's plenty of stable desktops to try in the meantime. KDE Plasma doesn't get enough attention, and likewise Bodhi Linux's Moksha. Also, more

people should run everything from a terminal under *i3*.



Michael Reed

I suppose one guy has to be Mr Boring. I start with a minimalist Xfce desktop. I add the Whisker menu for a searchable launcher, main panel at the bottom, and then I add CPU and memory displays to

the status area. Fast and reliable for getting work done.



Nate Drake

Aside from being a great *Scrabble* word, LXQt is perfect if you need a lightning-fast, no-frills responsive desktop. Not only do menus open in record time but it also comes bundled with a carefully curated selection of lightweight apps, such as

QTerminal. Did I mention it's also window-manager-agnostic?



Les Pounder

I started my Linux journey with Gnome, but then I moved over to KDE, which then matched my workflow. A few years later I joined the surge of Ubuntu users and went back to Gnome. But in recent years, I've

moved back to KDE; again, it just matches my workflow.



Nick Peers

Not a desktop per se, but the Cockpit web-based interface is a great way to access your distro remotely. It's got everything you need to check in from another machine on your network, including a fully functional terminal

and access to settings such as services, storage and networking.

*Savings are based on the cover price.

Lightning strikes



Linux isn't a magic bullet. It still has bugs, it still has crashes, it still can't stop user error (although *rm* is harder to run on root these days) and it still can't stop poorly tested updates. There seemed to be some smugness around the CrowdStrike incident (*read about that in news, page 6*) that took down certain Windows systems worldwide in July; the only thing was, a CrowdStrike update also

took down Red Hat and Debian systems in April.

You'd be right to point out that this is a third-party issue, so what's my point? Specialisation in the software world can lead to Bad Things™ happening. This corporate meltdown occurred because there's a single OS provider and a single antimalware vendor with, as it turns out, questionable quality control.

That's not to say the open source ecosystem doesn't have its own weaknesses and vulnerabilities in the way it's developed, but at least these are generally known and in the open – the XKCD cartoon of a pyramid of blocks all balancing on one tiny block (which is some obscure project maintained by a single person) at the bottom springs to mind. Most importantly, it provides an abundance of choice. We're looking at Linux Mint 22 this issue, the perfect green-debloating cure to those Ubuntu Snaps blues – or is that oranges? Choice is what sparked Linux Mint in the first place and to this day it's still what's driving its development and keeping users flocking to its comforting and enjoyable desktop experience!

Neil

Neil Mohr Editor
neil.mohr@futurenet.com



Subscribe
& save!

On digital and print:
see page 16

Contents



REVIEWS



HighPoint Rocket 1608A 19

Looks like we've found a storage solution for when our very own rocket man, **Shane Downing**, who really needs that 56GB/s.

Q4OS 5.5 20

Nate Drake is on cue to explore the latest Q4OS and comes away amazed at its speed and simple setup.



Pardus 23.2 21

Nate Drake offers a chunk of Turkish Delight as he tours this Debian-based distro, coded with government approval.

Nobara Project 40 22

Nate Drake delves into this modified Fedora distro. Is this the ultimate in Linux gaming or should you say "no Nobara"?

Mabox Linux 24.07 23

A spin of Manjaro, which is a spin of Arch, is leaving the normally centred **Nate Drake** in a right tizzy.



Men of War 2 24

Wondering what war is good for, **Jake Tucker** is reminded by The Management that it looks after his pay cheque.

FRESH MINT 22!

Linux Mint is officially in season and **Jonni Bidwell** has been sniffing it voraciously. Join him on a fragrant voyage of discovery. See page 32!



ROUNDUP



Remote desktops 26

Ever wish you had the desktop of another computer right in front of you? **Michael Reed** checks out five applications that can do this and more.

IN DEPTH



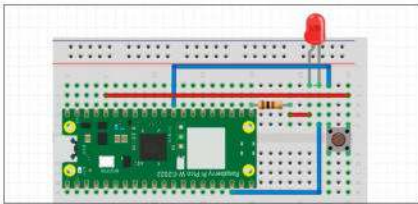
PipeWire 46

In the fourth instalment of our *Inside Linux* series, **Matt Holder** delves into the depths of our audio stack to learn about PipeWire and what came before it.

PI USER



Pi news 41
Les Pounder makes connections with the latest Pi product, plus more stories from the world of single-board computing.

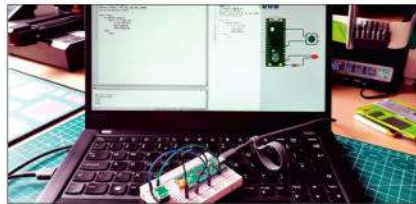


Fritzing 1.0.3 42
Les Pounder spends most of his work hours looking at circuit diagrams, then does the same thing after work.

CREDIT: OpenAI, Fritzing.



Sovol SV08 43
 Big, super-fast and open source, it looks as though **Denise Bertacchi** has found her perfect printer!



Simulate Pi projects with Wokwi 44
Les Pounder simulates an electronics project because he is too lazy to get up and build it for real.

CODING ACADEMY

Code your own Wordle-like game 90
 Keeping his grey matter well tested, **Matt Holder** creates a *Wordle*-style game and hopes the lawyers don't ask questions.

How to build a 6502 assembler 94
David Bolton shows how to create a 6502 assembler in C#, highlighting the problems you need to overcome.



REGULARS AT A GLANCE

News 6
 Linux is CrowdStruck; Nvidia opens up; Google eats cookie words; Jellyfin has firm funding base; Swiss roll out open source; ext4 accelerates; plus industry opinions and a look at new distro releases.

Kernel watch 10

Answers 11
Neil Bothwick claims he was born and raised on a server farm, which is why he finds it a breeze to answer questions about file transfers, missing folders, night-time monitoring, overflowing logs, and more.

Mailserver 14
 Readers assail **Neil Mohr** with queries about ugly fonts, Linux gaming and more.

Subscriptions 16
 Grab your monthly dose of Linux and save a massive 50% off the usual price!

Back issues 60
 Get hold of previous *Linux Format* editions.

Overseas subscriptions 61
 Get *Linux Format* shipped around the globe.

HotPicks 83
Mayank Sharma is grumpier than usual since he binge-watched *TDF: Unchained*, and was then a domestique to open source gems, such as *XFE*, *Taon Music Box*, *Puddletag*, *RetroArch*, *LibreWolf* and more, as he raced to deadline.

Next month 98

TUTORIALS

TERMINAL: Try a tiny text editor 50
Shashank Sharma lives by the try-it-before-you-dismiss-it code. That's how he finds new tools to replace his defaults.

LINUX BASICS: The filesystem 52
Nick Peers explains the basics of using the Linux filesystem, from managing your files to learning about permissions.

NIXOS: Geek out easily 56
Mats Tage Axelsson shows how easy it is to use NixOS and the challenges of installing new apps. All 100,000 packages.



RAID: Expand your storage 62
 Although he likes to think he lives life in the fast lane, **Neil Mohr** isn't sure what to do with 6TB of 3GB/s storage...

OLLAMA: Control multiple AI models 68
 Taking his tentative first step towards world domination, **Tam Hanna** discovers how to control all the AI LLMs he needs.

STEAM DECK: Record games 72
 Not-at-all self-obsessed **Dave Meikleham** can't stop watching his own gaming skills thanks to his Steam Deck's new feature.

ADMINISTERIA

Administeria 74
 Ever the busy boy, **Stuart Burns** needs things to run on time, so he's here to show you how to set up schedules.

AngelVPN 76
 Always a good boy, **Sam Dawson** thinks he's rather smitten with this devilishly good service.

PureDome VPN 77
 A VPN service that's aimed at business types like **Mike Williams**, who prefers to do his private business in private.

Whatever happened to Symbian? 78
 The Symbian OS had a lot going for it, and even dabbled with open source, so why did it fail? **Mike Bedford** investigates.

Newsdesk

THIS ISSUE: Linux is CrowdStrike » Nvidia opens up » Google eats cookie words » Jellyfin's firm funding base » Swiss roll out open source

IT OUTAGE

CrowdStrike issues hit Microsoft and Linux

Microsoft wasn't the only victim of faulty CrowdStrike updates. Weeks before the worldwide outage, Debian and RHEL were also hit.

As most of the world knows, on 19th July 2024, a faulty update was released for CrowdStrike's *Falcon* sensor. This caused an out-of-bounds memory read in the Windows sensor client, resulting in millions of devices entering a boot loop.

Although a fix was quickly released, this had to be applied manually on each affected machine, meaning that at the time of writing, almost a week later, the effects of the biggest outage in IT history are still being felt. According to CNN, insurers have calculated that the total damage from this outage to Fortune 500 companies is over \$5.4 billion.

This was a good time for organisations to be based in China, where companies tend to avoid US software like *Falcon*. Russia and Iran, which are subject to embargoes on US programs such as *Falcon*, also reported no major outages.

Readers may be tempted to think this was also a good time to feel smug about being a Linux user. While it's true that the outage on 19th July didn't directly affect Mac OS or Linux hosts, CrowdStrike was actually responsible for separate outages weeks before the major event occurred.

In May 2024, Rocky Linux user Antoine posted in the OS forums. He'd encountered an issue with upgrading to Rocky Linux 9.4 on a server equipped with *CrowdStrike Falcon*, which resulted in a system freeze due to kernel panic.

Given that Rocky is based on Red Hat source code, it wasn't surprising that the company also posted multiple warnings on its own forums

about a similar issue affecting RHEL. Not only did RHEL experience the same kernel panic but *Falcon* could also cause system crashes on both RHEL 6 and 7. Suggested workarounds included contacting CrowdStrike for support or disabling *Falcon* altogether.

It seems Debian users haven't been spared either. Posting in Hacker News, user JackC stated that his Linux production fleet had experienced errors like this in mid-April.

He explained that his organisation (a civic tech lab) runs a bunch of different websites on various architecture, all of which were protected by CrowdStrike. According to JackC, the company released a software update that was incompatible with Debian stable. After a few days it caused "multiple websites and cloud servers to simultaneously hard crash and refuse to boot".

Crucially, the machines ran perfectly when the security software was removed, although CrowdStrike apparently demanded more proof and took several days to acknowledge the bug.

On the plus side, *systemd* lead developer Lennart Poettering has pointed out how these situations can be averted on Linux systems by leveraging *systemd*'s Automatic Boot Assessment functionality.



A digital sign at Dulles International Airport, Virginia, displaying the Windows BSOD (blue screen of death) during the incident.

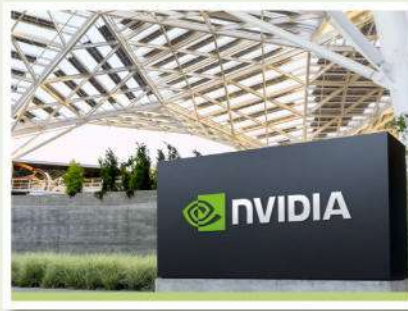


Rocky Linux user Antoine noticed upgrading servers running Falcon caused kernel panic. He also spotted RHEL had the same issue.

GRAPHICS

Nvidia officially goes open source

The R560 driver marks Nvidia's full transition to open source GPU kernel modules.



Nvidia has now completed the process of open sourcing GPU kernel modules with the R560, but older GPUs aren't compatible.

Nvidia has something of a love-hate relationship with the open source community. The company has often refused to provide source code for drivers to allow Linux operating systems to take full advantage of all features.

This conflict reached its nadir in 2012, when in a video interview, Linus Torvalds called Nvidia "the single worst company" he and his associates had ever worked with, followed by giving them the middle finger.

Two years ago, ransomware gang Lapsus\$ took matters into its own hands by threatening to leak GPU data files unless the company agreed to fully open source all of its drivers in perpetuity.

While the chip manufacturer refused to give in, in 2022 it did begin open sourcing GPU kernel modules, starting with the R515 driver for data centre GPUs, using dual MIT/GPL licensing.

Nvidia has now announced that it intends to do the same for the R560 driver, the final stage

in open sourcing all of the company's GPU kernel modules.

In most cases, the default driver installed will now be the open source driver instead of the proprietary one. As of CUDA 12.6, the top-level CUDA package will now include the open source driver. This is an improvement, as previously users had to install the Nvidia driver open package separately from the CUDA Toolkit.

The company's Tech Blog points out that said modules won't be compatible with all chips. This includes older GPUs, such as those that deploy the Maxwell, Pascal, or Volta architectures. In these cases, the blog recommends using the proprietary driver. Nvidia also recommends doing the same for mixed deployments, where older and newer GPUs are used in the same system.

OPINION

SUITE PRIVACY



Italo Vignoli is one of the founders of LibreOffice and the Document Foundation.

Think about the personal and confidential information in your office suite documents; it's essential your office suite respects user privacy. LibreOffice does not ask you for any personal information in order to use it – the email address requested to make the donation is only linked to a credit card or PayPal.

LibreOffice is the only office suite that truly respects your privacy, thus ensuring that you can decide if and with whom to share content you have created.

The issue of privacy, often framed within the broader issue of digital sovereignty – the ability to control and manage one's profile and digital content – is becoming increasingly important. Many users are starting to worry about the apparent use of their data, including what they thought was private, by large tech companies – especially in the US – for commercial purposes, with the cross-fertilisation between those who take the information and those who use it being a real concern.

If I mention a trip to Croatia in a document, would I then expect to start receiving adverts about the possibility of cheap dental treatment in Zagreb and the surrounding area? Using LibreOffice for content management, this would never happen.

WEB BROWSER

Google backs down on cookies

Chrome will allow users to make "informed choices".

Given the huge market share owned by Google and its flagship browser Chrome, it's understandable why some advertisers were alarmed in 2020 when the tech giant announced its plans to phase out third-party cookies.

Originally, this was part of a two-year timeline for Google's open source Privacy Sandbox project, but this was quickly derailed. In July 2022, the deadline to eliminate all tracking cookies was moved to the end of 2024, ostensibly to give advertisers more time to adjust to the Sandbox APIs.

More cynical commentators pointed out that the delay also may have been due to Google's plans being subject to scrutiny by the UK's CMA (Competition and Markets

Authority), as well as the US DOJ (Department of Justice).

In April, Google announced the deadline to implement Privacy Sandbox had been pushed back again – to 2025. During this time, Google had experimentally introduced a new feature called Tracking Protection, which blocked third-party cookies for 1% of Chrome users. Its blog said it wanted more time to allow the CMA to check Privacy Sandbox wasn't anti-competitive.

In July, Google VP Anthony Chavez announced that the software will now follow a "new path". This will involve halting plans to deprecate third-party cookies. Instead, Chrome users will be offered a chance to configure their settings to make an "informed choice" that applies across their web browsing.

OPINION

EARLY ADOPTER



David Stokes
is a technology evangelist
at Percona.

“The logic behind MySQL upgrades used to be simple. You want something near current, as a cutting-edge release of any server software can introduce incompatible changes to your data. But you can't lag too far behind or you miss out on necessary bug fixes and new features.

Today, many teams are choosing not to move at all. Why? Performance. MySQL 5.7 is significantly faster than MySQL 8.0. Early reports on MySQL 8.4 say this trend continues. Metrics collected by our *Percona Monitoring and Management* tool report that many systems still run on MySQL 5.7 nearly a year after reaching end of life.

So, what are your options? Oracle offers MySQL Community Server in versions 8.0.38, 8.4.1 Long Term Support and 9.0. V9.0 is an innovation release, not for production. Oracle promises minimal changes in the 8.4 series, providing only necessary updates to ensure stability. There are also Enterprise editions for those with deep pockets.

These alternatives may not be attractive. So, what can you do? This is the beauty of open source – you can look at other builds that provide enterprise features and cover performance needs, too. You can also get extended support for 5.7. Open source keeps your options open. ”

FUNDING

Jellyfish devs say help others

Lead developer Joshua Boniface has asked users to support clients instead.

Most Linux users are familiar with the meme depicting an elaborate machine representing all modern digital infrastructure. The machine is propped up by a tiny component labelled, “A project some random person in Nebraska has been tirelessly maintaining since 2003.” As such, it's easy to believe that all open source projects are underfunded and that pro developers can only spare the odd coffee break to work on them.

This doesn't seem to be the case for FOSS media server Jellyfin, however. In early June,



Unlike many FOSS projects, Jellyfish is well funded for the next few years. Users are being asked to support clients instead.

lead developer Joshua Boniface created a post in the user forums entitled “We're good, seriously.”

He went on to explain that the project has around \$24,000. Given that current expenses are around \$600 a month, he calculated that this is sufficient to keep Jellyfin going for more than three years.

Boniface also described client support as the “hardest part of the Jellyfin ecosystem to keep going”. He then asked users to donate to the authors of clients for the time being.

CREDIT: Jellyfin Main Page

SOFTWARE

Switzerland is open source

New law requires all government apps to be open source.

Switzerland has enacted the Federal Law on the Use of Electronic Means for the Fulfilment of Governmental Tasks (EMBAG), which in theory requires use of open source software for all sector bodies.

A closer inspection of Article 9 reveals that federal authorities subject to this law must publish the source code of any software they develop, unless this clashes with third-party rights or important security concerns.

The European Commission has stated that the implementation of EMBAG is expected to serve as a model for other countries considering similar measures.

Swiss authorities must publish the source code of software they develop – but not necessarily under an open source licence.



FILESYSTEM

Ext4 gets faster

The latest Linux 6.11 comes with performance optimisation for ext4.

Ext4 was first released back in 2006 with Linux 2.6.19 and has become the default filesystem for many popular distros.

One of ext4's original developers, Ted T'so, posted recently in the Linux Kernel Mailing List to discuss ext4 updates for Linux 6.11: “Some performance improvements; in particular, improving IOPS and throughput on fast devices running async direct I/O by up to 20% by optimising `jbd2_transaction_committed()`.”

Thanks to these JBD2 optimisations, users have indeed reported performance improvements of up to 20% on devices using async direct I/O.



The latest ext4 contains “cleanups and bug fixes”, especially for the fast commit feature. IOPS has also been boosted.

Distro watch

What's behind the free software sofa?

UBUNTU*PACK 24.04

This Ubuntu remix is the creation of Ukrainian developer UALinux, an official Canonical partner. The OS comes in several editions, including Desktop, Education, Game, Server and Rescue. It contains more drivers and media codecs than stock Ubuntu and the latest release supports 10 desktop environments, including LikeWin, a Gnome variant. Unlike Ubuntu, the default browser (*Firefox*) and mail client (*Thunderbird*) have been unlinked from Snap. See <https://ualinux.com/en/ubuntu-oem>.



Ubuntu*Pack offers superior support for various codecs.

MIDNIGHTBSD 3.2

This distro is derived from FreeBSD, with an emphasis on creating an intuitive desktop environment with graphical ports management. This year, the OS celebrates its 17th birthday. The latest release (3.2) introduces the Ravenports universal package system for AMD64. You can choose to bootstrap this during install. If you do so, you can then install software packages using `/raven/sbin/ravensw`. The developers warn against mixing imports and Ravenports packages. Visit <https://www.midnightbsd.org/notes/> to read the full release notes.



V3.2 introduces Ravenports for installing packages.

QUBES OS 4.2.2

Qubes is to security-oriented OSes what Fort Knox is to a piggy bank. It uses Xen-based virtualisation to create multiple qubes (VMs), which function as isolated compartments, separating out your personal data. Naturally, it's sometimes necessary to move content across qubes. The latest version contains a fix for a restriction that would prevent copying files containing unusual unicode characters, striking a better balance between security and usability. Discover more at www.qubes-os.org.



The OS manages multiple VMs to isolate your personal data.

OPENMANDRIVA 24.07 ROME

This operating system started its life in 2013 as a fork of Russian Linux distro ROSA, which itself was derived from Mandriva. Last year, the project developers introduced a rolling branch code-named ROME. The latest version now uses the KDE Plasma 6 desktop by default. There are also spins featuring LXQt (2.0.0 Qt6) and Gnome (46.3). A Wayland ISO is also now available. Users can even now install *Proton* as an OpenMandriva package. You can read more about the distro at www.openmandriva.org.



OpenMandriva has an updated desktop and support for Proton.

CLONEZILLA LIVE 3.1.3-11

Clonezilla isn't a traditional Linux desktop distro but is instead designed to be run as a live DVD. The OS's main purpose is for partitioning and cloning disks. This release is based on the latest Debian Sid repository and contains an updated Linux kernel (6.9.7-1) as well as an upgraded version of the *Partclone* utility. There's also better support for German and Spanish languages. A number of faulty packages have been removed. Learn more at <https://clonezilla.org>.



Clonezilla is designed for partition/disk imaging and cloning.

OPINION

ATOMIC STEAM



Ludovico de Nittis is a software engineer at Collabora.

Since the Steam Deck's release back in 2022, users have had a portable means to enjoy Linux-based gaming. As with any system that advances, there have been many under-the-hood improvements.

A recent example involved atomic updates and the shift from *Casync* to *Desync*, a tool offering faster system updates, parallel HTTP requests, and better error handling. With *Desync*, it is now significantly faster to apply system updates. How much faster depends heavily, of course, on the size of the update and the internet connection speed. From our testings, the overall update process was consistently at least 30%-50% faster.

Also, if the download of an update fails – if there is a connection error, for example – the second download attempt can quickly resume from the point previously reached. The download progress percentage is also more precise.

Lastly, *Desync* allows us to quickly check if the current image is pristine or has been altered. This can be very useful, for example, during a factory reset to avoid having to download an image unless necessary.

Want to give this a try? The change is included in SteamOS 3.6, in the Preview channel. Opt in from Settings > System > Steam Update Channel.

OPINION

FILTER
TIPS

Jon Masters is a kernel hacker who's been involved with Linux for over 22 years, and works on energy-efficient Arm servers.

“The main topic of conversation in the computing industry over the past few weeks has been the giant worldwide outage apparently caused by a bad update from cybersecurity company CrowdStrike.

The crux of the issue is that cybersecurity firms have sold the world on the need to install proprietary kernel drivers that hook into the guts of the OS in a manner similar to antivirus software. These drivers are invasive, and when they go wrong, can cause a kernel crash regardless of OS. The fix, in my mind, is to get out of the business of installing third-party kernel drivers on production machines, especially if they are to receive live updates from an independent third party that could be disruptive.

The need to check the compliance box as well as general unease is going to doom any attempt to stop using such software and I'm not proposing that. But I am glad to see a blog post from Brendan Gregg (<https://bit.ly/lxf319friday>) about Windows adopting the eBPF tech widely deployed by Linux. eBPF allows filters to be loaded into a kernel. These can allow the same kinds of behaviour as an antivirus/security driver without directly modifying the kernel code.



Kernel Watch

Jon Masters summarises the latest happenings in the Linux kernel, so that you don't have to.

Linus Torvalds announced the release of Linux 6.10. The new kernel includes features such as a new mseat system call that protects a virtual memory range against modifications, so you can mseat memory to prevent accidental or malicious – as in a buffer

cent or much more in some games) the overhead of emulators such as *Wine* and has shown some significant frame rate improvements on a number of Windows game benchmarks.

With the release of Linux 6.10 came the opening of the merge window (the period

of time during which disruptive changes are allowed into the kernel) for what will be Linux 6.11 in a couple of months. Features already merged into 6.11 include the ability to run as a guest under AMD's Confidential Computing architecture (known as SEV-SNP), and a vDSO (Virtual Dynamic Shared Object) implementation of the

getrandom system call. System calls have overhead, and the vDSO is a small region of memory inside each process where calls have the overhead of a local function call. Thus, this new implementation should speed up random number generation. **LXF**

OVERHEAD BLOCKER

“This feature reduces (dramatically – tens of per cent or much more in some games) the overhead of emulators such as *Wine* and has shown frame rate improvements.”

overflow type scenario – changes to permissions.

The new kernel also includes a new ntsync driver that implements Windows semantics for thread synchronisation. This feature reduces (dramatically – tens of per

» ONGOING DEVELOPMENT

In what seems like a monthly occurrence, additional machines that this author greatly enjoyed several decades ago are on the chopping block as far as Linux support is concerned. Arnd Bergmann posted a thread titled “[RFC] Arm architecture board/feature deprecation timeline”, which aims to remove a large number of legacy platforms from the pre-DT (Device Tree, a means of describing hardware in embedded systems) era of Arm v3, 4, 5 and even some Arm v6. Arm v7 remains a popular (but legacy) 32-bit architecture, while Arm v8 and v9 are current. In that sense, removing decades-old stuff does make sense, but the inclusion of the original StrongArm SA1100, as seen in original pocket PCs, such as the Sharp Zaurus, hits close to home.

Further patches have been posted for the new Qualcomm X Elite Arm-based laptops but there has been some controversy and

pushback from kernel folk who observe that graphics firmware blobs required to boot must be per-OEM signed and thus can't easily be shipped in an upstream kernel. The author has one of these laptops and will walk through the latest patches in a future issue, after playing a little more with the state of development.

Rust support continues apace. This month saw patches posted to enable the kernel's modversions mechanism, needed to load drivers that ship separately from the kernel. There were also patches posted enabling support for running Rust code in an interrupt disabled context.

RISC-V patches underway include support for running under nested virtualisation in which SBI (Supervisor Binary Interface, aka firmware running in machine mode) calls can skip a level of trap-and-emulate otherwise required.

Answers



Neil Bothwick
claims he was born and raised on a server farm.

Got a burning question about open source or the kernel?
Whatever your level, email it to answers@linuxformat.com

Q Cordless phone

I need an easy way to transfer files between my Linux PC and my Android phone, a Samsung S21 Ultra. These are mainly music and image files. I have tried a few different ways (particularly via USB cable) and had some limited success. However, although USB cable should be the easiest, I'm finding it awkward and prone to not mount the handset or drop out after a short amount of time.

Is there an easy way to transfer files wirelessly on the same network? If so, what are the best options or apps?

Abigail Nolan

A

Android phones use MTP (Media Transfer Protocol) for transferring data over USB. There's a number of MTP filesystems for Linux, such as `jmtpfs`, that can mount an MTP device for file copying, but none seems startlingly reliable.

The simplest way to transfer smaller files is to copy them to your Google Drive account, which you have if you use Android, then you can access them on your computer and your phone. Native Drive support in Linux is poor, but you can use a web browser to transfer files. It's how this text arrived at LXF Towers.

Another option is *AirDroid*, available from the Play Store. It allows a lot more

than file transfers but, once again, Linux users are restricted to the web client. *AirDroid* is a paid program, with the free version restricted to 200MB per month, making it only suitable for very light or heavy users.

If you want unrestricted copying, you can mount your phone over the network using SSHFS. First you need to install *SSH Helper* from the Play Store and set a password. You may also need to install SSHFS on your computer. Now you can mount your phone's storage by making a note of the IP address shown in *SSH Helper* and running the following command in a terminal on your computer, using the IP address you just noted:

```
$ sshfs IP-ADDRESS:/storage/emulated/0 ~/Android -p 2222
```

The mount point, `~/Android` here, must exist and be empty.

Another suitable alternative is *Syncthing*, available for Android and Linux (and just about everything else). This program provides automatic synchronisation between shared folders on devices, so it is simply a matter of copying a file to the shared folder, then it automatically appears on the other device(s). *Syncthing* is configured through a web interface, but this the intended way, not a poor man's alternative. Once set

up, the range of options is large, with any combination of multiple shared directories on multiple machines being possible.

Q Hide and seek

I was given a computer on which I installed Linux Mint and I love it. I booted the machine today – Mint is running fine but the folders of files that I had created are all gone. It is as though the OS reset itself to a fresh install. The motherboard has a loose connection for the CMOS battery – nothing a bit of duct tape couldn't fix, but before the tape, I had to reset the BIOS. Mint then booted fine but with the file loss as mentioned.

Is there any explanation for this situation? Is this a coincidence?

Sam Walsh

A

Files do not disappear for no reason, and that reason is usually a drive failure. How is the machine set up? Are the missing folders on the same partition/drive as the root filesystem? If not, are they all mounted? The first thing you should do is check the system log for any errors – as you are using *Systemd*, the `journalctl` command will do this for you:

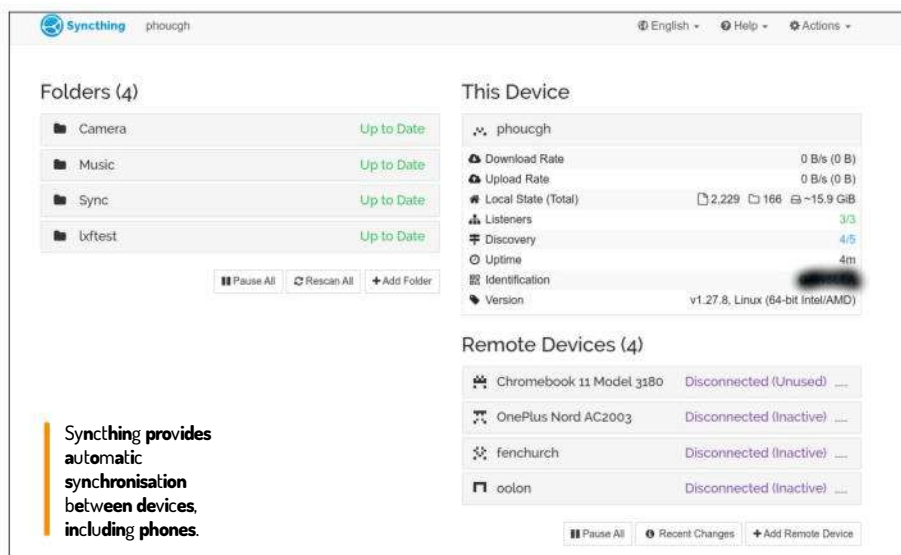
```
$ sudo journalctl -b -p err
```

This shows all journal entries since the last boot with a log level of error or higher. If this does not report anything helpful, try:

```
$ sudo journalctl -b
```

This provides a lot more output – everything since the last boot – but `journalctl` pipes the output through the less pager to make it more manageable. You can press Space to move through a page at a time, or press / followed by some text to search for that text. We suggest searching for 'mount' to detect any mount errors. Once you have an error to deal with, you can find the solution. Unless the drive containing those files has expired, they are still there – it is just a matter of finding them.

It may look like a coincidence, but you may have disturbed something while duct-taping your motherboard. You should check that all cables are securely connected to the motherboard, preferably



by unplugging and replugging, because this can also help with dirty connections.

You could also try using the `find` command to see if one of the files is visible to the operating system. Pick one file that you know the name of and run:

```
$ sudo find / -name "name of file"
```

We use `sudo` here to ensure the file command is able to traverse the full filesystem, irrespective of any permissions that are set.

Q Silent night

I use *monit* to keep an eye on various systems, but I do not want it to disturb me at night. So, I have set up a couple of *cron* jobs. In the morning: `systemctl start monit`; while at night I use: `systemctl stop monit`.

Is this the best way to do it or do you have an alternative?

Corey O'Neill

A You could do it this way, but it is a bit of a blunt instrument. You are stopping all monitoring at night, which may or may not be what you want. Also, there is always the possibility of one of the *cron* tasks failing for any number of reasons, leaving you either with no monitoring or monitoring all night. *Monit* itself has the ability to schedule when monitoring runs, and this can be set separately on each service. This means that, for example, you can disable any noisy services – those that pester you with messages when something goes wrong – while still running those that silently restart a process when it fails. It is also useful if you have some processes or machines that are only available at certain times, so you avoid unnecessary alerts from something that you know should not be available.

All of this is done with the **every** statement, which you may already be familiar with because you can use it

System status	
Parameter	Value
Name	phoucgh
Status	OK
Monitoring status	Monitored
Monitoring mode	active
On reboot	start
Load average	[0.52] [0.31] [0.26]
Cpu	2.2%usr 1.0%sys 0.0%nice 0.1%iowait 0.0%hardirq 0.0%softirq 0.0%steal 0.0%guest 0.0%guestnice
Memory usage	2.8 GB [9.0%]
Swap usage	461.8 MB [5.6%]
Uptime	6d 16h 3m
Boot time	Fri, 28 Jun 2024 19:57:33
Filedescriptors	10816 [0.0% of 9223372036854775807 limit]
Data collected	Fri, 05 Jul 2024 12:01:03

Monit can keep an eye on your system at all times, or only when you ask it to.

to reduce the frequency with which a process is checked, for example:

```
service foo with pid /var/run/foo.pid
every 6 cycles
```

If you have *monit* set to check every 30 seconds, **foo** is only checked every three minutes. However, **every** can also accept *cron* specifications. For example, the following checks the service between 9am and 5pm, Monday to Friday. This sounds like it should do what you need and gives fine control per service. Note that the *cron* specification is in quotes.

```
service foo with pid /var/run/foo.pid
every "* * 9-17 * * 1-5"
```

You could also use:

```
service foo with pid /var/run/foo.pid
not every "* * 0-8,19-23 * * "
```

This works in the same way except the *cron* specification says when the service should not be checked – in this example between 7pm and 8am every day. It doesn't matter which of the two you use; pick the one that gives the most readable configuration for when you look at it again in six months' time.

Q Overflowing logs

I am using Debian 12. Recently, I discovered that my 30GB root partition is 91% occupied. What could be the causes and how do I discover them?

My root **home** directory is empty. I have recently installed *Apache Tomcat* server and *Eclipse Web IDE*, but both are in my **home** directory, which is in a separate partition. My **/var/log/journal/** directory contents occupy 2.7GB. Is it OK to delete files contained in **/var/log**?

Daniel Forster

A Servers often store their files under **/var/lib**, so even if the *Apache Tomcat* executable is in your **home** directory, check its configuration to see where it keeps its data files. A 30GB root partition should be plenty large enough for an operating system, but fills up if you begin storing data in there, too. Your **/root** directory is probably not empty because some processes store files in there, but they could all be in dot files – files (or directories) whose name starts with a dot

» A QUICK REFERENCE TO... SSHFS

If you need to access files on a remote computer, you can use the normal file-sharing methods, such as NFS and Samba, but this needs root access to the remote machine and you have to spend time configuring it. If you have SSH access, even as a non-root user, you can copy across files using the standard terminal commands and copy them with `scp`. It

would be nice to have the convenience of file sharing and the simplicity of an SSH connection. That is where SSHFS comes in, it is a FUSE filesystem that can mount remote paths using only SSH. The basic command is

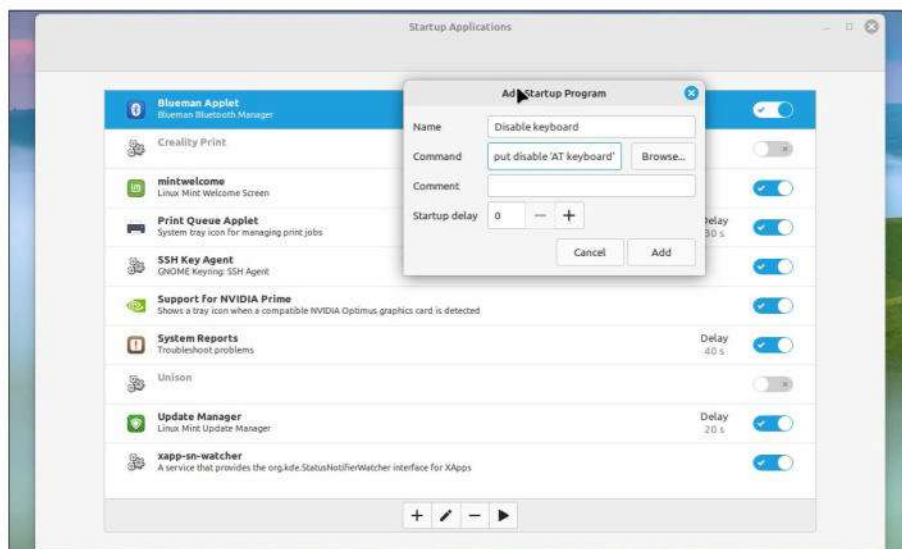
```
$ sshfs user@remote_
host:remote_dir mount_
point
```

As with standard SSH connections, you can omit

the username and it uses your local user. If you do not specify the remote directory, SSHFS mounts the remote user's home directory. A password is requested unless you have set up key authentication, the same as with SSH. The mount is only available to the user who mounted it. To make it available to others, add `-o allow_other -o default_`

permissions. There are other options, all documented in the man page. If your SSH server is running on a non-standard port, you can give that with the `-p` option.

If you want to mount an SSHFS filesystem at boot, add it to **/etc/fstab**, set the filesystem type to SSHFS, but make sure the remote host is available if you don't want the mount to hang.



❗ Mint's Cinnamon desktop, like most, has options for you to run whatever commands you need when it starts up.

and are hidden when viewed in a file manager, or even with `ls` unless you use the `-a` or `-A` option. However, these files are generally small and are unlikely to be the source of your problem.

Your `/var/log/journal` is only using 10% of the filesystem size, which is the default. So reducing that will not have a significant impact, especially if something else is filling the filesystem. You shouldn't delete files from the `Systemd` journal manually but it does provide a means to reduce the space used in a system-friendly manner. One of the following may help:

```
$ journalctl --vacuum-time=14d
```

```
$ journalctl --vacuum-size=1G
```

The first removes all archived journal files older than the specified time – two weeks in this example – while the second removes enough files to bring usage down below the given size. In this example, 1GB is the usage you want to achieve; the command would remove around 1.7GB.

It is also worth checking `/var/log` for large files and possibly installing `logrotate`, if you do not have it. This reduces usage by starting a new version of each log file set in its configuration, then compressing the old one. As log files contain a lot of repetitive text, this can reduce space usage for those files by 90%. The default setup is to keep the last four archived logs and delete older ones, so you always have at least four weeks of data.

One program we find extremely useful for finding out where all your disk space has gone is `ncdu`. This is not installed by default on most distros but is in the repositories. It is a terminal program, so is ideal for running as the root user without a GUI, or even over SSH. Simply run:

```
$ sudo ncdu -x /
```

The `-x` switch stops it descending into other filesystems mounted under `/`, so it

gives a true picture. Directories are listed in order of size and you can descend through them to find the culprits. If you have any doubts as to whether a file is needed, rename it rather than delete it. If your system continues to run OK, you can delete it later. Of course, you should have a recent backup before you start deleting potentially important files.

Q Random keys

I have an older Toshiba Satellite laptop on to which I've installed the lightweight Linux Mint 21.2, but the keyboard is playing up, typing long lines of characters randomly. I've tried cleaning the keyboard but no luck. I've been trying to disable the keyboard in Linux and just plug an external USB keyboard in.

Once it's booted up, I can do this in a terminal by typing the following, but if it's already playing silly beggars with its repeated characters, it's difficult, even if I just press the up cursor to access the previous time I executed the command:

```
$ xinput disable "AT Translated Set 2 keyboard"
```

I have been trying to put the command in the startup scripts so it does it while loading. I found a suggestion I should put a script into `/etc/init.d` that contains:

```
#!/bin/bash
id=$(xinput list --id-only 'AT Translated Set 2 keyboard')
xinput float $id
```

It doesn't work. Once Linux is loaded, the internal keyboard still works. I have made sure the script is executable – can you see a reason why it won't work? Is the `init.d` folder correct for Linux Mint 21.2?

Joel Kaur

A Have you checked whether the BIOS has an option to disable the

internal keyboard? If not, there's a number of other ways to achieve this. Using `xinput` works, once X is running, so running it from `/etc/init.d` may not work. Linux Mint uses `Systemd` and that does not use `/etc/init.d`, although `Systemd` does usually execute the contents of `/etc/rc.local` for backwards compatibility. The simplest way to run `xinput` is from your desktop's startup mechanism. Put the following in `~/.xinitrc` or put it in a script and call that from your desktop's startup options:

```
xinput disable "AT Translated Set 2 keyboard"
```

Using `xinput disable` is preferable to using `float` as it is easier to re-enable the keyboard should you need to. Also, the `float` options are deprecated in `xinput` and may cease to work in a future release.

You can disable a laptop's keyboard earlier in the boot process, provided it uses the `i8042` keyboard driver. When the boot menu appears, press `e` to edit the current entry and add the following:

```
i8042.nokbd
```

If this works, it disables the keyboard right after the boot menu, not just for the desktop. To make this permanent, add the option to the `GRUB_CMDLINE_LINUX_DEFAULT` in `/etc/default/grub` and run:

```
$ sudo update-grub
```

Now the keyboard is only active until the boot menu. To disable it even further, we need a hardware approach. If you feel comfortable, you could open the case and disconnect the ribbon cable from the keyboard. Search YouTube for a video showing how to open your model, but only do this if you are comfortable doing so. **LXF**

GET HELP NOW!

We'd love to try to answer any questions you send to answers@linuxformat.com, no matter what the level. We've all been stuck before, so don't be shy. However, we're only human (although many suspect Neil is ChatGPT v1), so it's important that you include as much information as you can. If something works on one distro but not another, tell us. If you get an error message, please tell us the exact message and precisely what you did to invoke it.

If you have, or suspect, a hardware problem, let us know about the hardware. Consider installing `hardinfo` or `lshw`. These programs list the hardware on your machine, so send us their output. If you're unwilling, or unable, to install these, run the following commands in a root terminal and send us the `system.txt` file, too:

```
uname -a > system.txt
lspci >> system.txt
lspci -vv >> system.txt
```

Mailserver

WRITE TO US

Do you have a burning Linux-related issue that you want to discuss? Write to us at [Linux Format](mailto:linuxformat.com), Future Publishing, Quay House, The Ambury, Bath, BA1 1UA or email letters@linuxformat.com.

Back to font

Having switched over to Linux Mint from Windows, I was a bit surprised by how bad some websites look – I'm using *Vivaldi* and *Firefox*.

I thought the web had moved to web pages that can display their own fonts, even if they're not installed on your computer. It seems that, for some reason, browsers in Linux continue with their ugly standard serif font, even if I load the same page in Windows using the same browser. It seems Windows runs the Verdana font but in Linux it reverts to whatever serif it is, which looks awful.

Mathew Tanner

Neil says...

I'm not an expert at web design and CSS ins and outs, but I think this is an interesting outcome of designing for the most-accessed platform, or just the platform the designer is using – likely Windows or Mac OS. Also, most Linux distros only ship with open source fonts. There's a Microsoft web font pack you can grab (here's one guide: <https://itsfoss.com/install-microsoft-fonts-ubuntu>) that might solve the default font looking odd. The Google open source replacements do tend to look indistinguishable, but that's not to say any distros include those by default.

A different-looking font can make a website look entirely different.



Linux optimisations make Nvidia hardware run as fast as possible.

Game matters

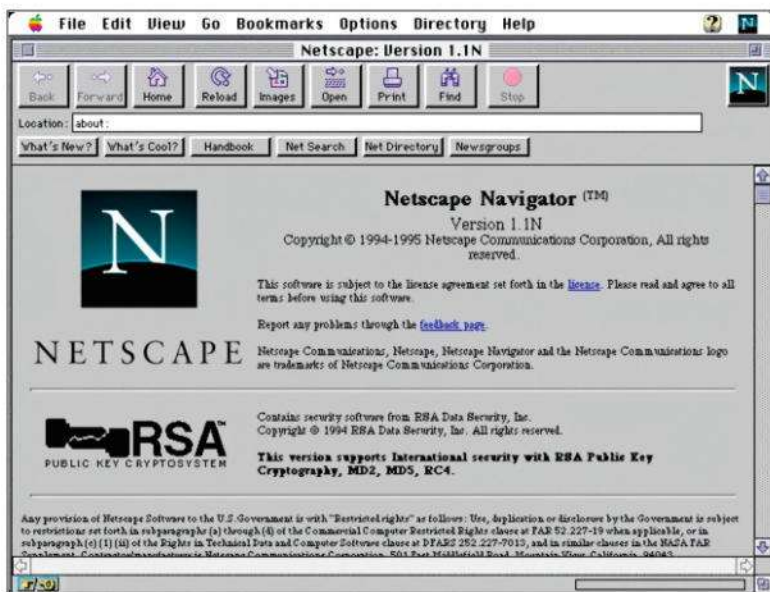
I was wondering if graphic cards give the same performance on Linux compared to Windows? As long as the game works on it through *Proton*, *Wine* or even runs natively, do we get the same performance?

Tim Hicks

Neil says...

There shouldn't be any performance difference between platforms running the same hardware, but it seems there is, at least on occasion. On Windows, gaming software uses DirectX as the layer between the game and the graphics card hardware. If that same Windows game is run on Linux using *Proton* (the Valve fork of *Wine* for Steam and the Steam Deck), those DirectX calls are translated into Vulkan rendering calls (and any other Windows system calls are translated to Linux ones) that work with Linux graphics drivers.

It seems on occasions the same game runs faster on Linux. Whether this is to do with underlying optimisations with Linux, hardware performance settings restricting Windows, or Vulkan being more efficient, no one's come out with a definitive answer, but the result can be more frames per second on Linux. Recently, open source benchmarks also put Ubuntu 24.04 ahead of Windows 11 for general productivity tasks. Window advocates pointed to efficiency settings in Windows holding back performance but I believe everything was set to defaults, which sounds like an even playing field to me.

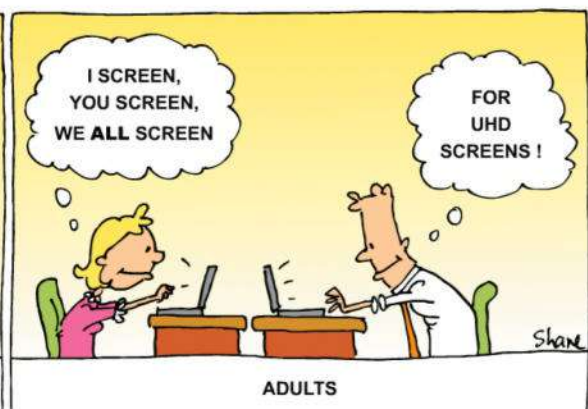


CREDIT: www.versionmuseum.com

Helpdex



CHILDREN



ADULTS

Licence to thrill

How does Android have an MIT licence when Linux is GPLed? I don't know much about licensing but my professor has told me that GPLed software forces other software to be GPLed, so if Android is based on Linux, Android should also be GPLed and any forks of Android should also be GPLed, making all of the forks by manufacturers and Android itself free software! Does that make sense?

Andrew Glitter

Neil says...

I'd get a new professor as that's a critically wrong interpretation of the GPL licence. But you also need to understand how large projects like Android are organised, which is to say that individual software contributions to a project can have their own specific licence. So, Android (it actually uses the Apache 2.0 not MIT licence) makes use of a multitude of separate projects, one of which is the Linux kernel, which is distributed under the GPL v2.0, but it's not derived from the Linux kernel project, so isn't bound by it.

It's actually more complex than this and there are entire software systems developed to track licence use and contributor copyright across large projects. So, the Android project does actually write Linux kernel contributions and these will be GPL licensed. On a larger level, the Android project is making use of the kernel in a compliant way, just as closed source kernel drivers use GPL-compliant shims to stop being contaminated by the GPL when distributed with the kernel, as they're loaded at runtime. I think... **LXF**



Android has been a weird Linux project for the longest of times due to how hardware support has to work on ARM devices.

» LETTER OF THE MONTH

Pi matters

Would a USB DVD/RW work with a Raspberry Pi 5? Could you look at the Checkmate 1500 Mini case for the Raspberry Pi 5?

Ian Learmonth

Neil says...

The short answer is yes, but there is a longer answer with regard to exactly what you want to do. The Linux kernel has all the drivers built in to automatically recognise and mount an optical drive over a USB bus. Usually it appears under `/dev/sr0` or on your desktop. Ensure you have the official power supply because using a bus-powered drive will certainly cause power issues otherwise. I've read about people having different levels of success with the Pi, but I'm assuming it's mostly down to bus power issues.

You should be able to just read files on any discs inserted; if you want to play DVD video, VLC has all the capabilities. I recently wanted to rip and stream an old DVD video and used *MKVToolnix*, but I believe *Handbrake* can be convinced to do this all in one with the addition of *libdvdcss*.

For straight DVD+/-RW use, again, yes, I believe most distros will treat an inserted rewritable disc as a hard drive (under `/dev/sr0`), but this will provide absolutely miserable write performance and probably sound as though the drive is being slaughtered. If you want to write to optical media, I would recommend using dedicated write software.

The Checkmate Amiga A1500 case does look fun. I'll see what we can do – Les might be interested in doing a build with that thing...



Neil has to keep reminding himself that his first Amiga's hard drive was just 20MB.

CREDIT: www.checkmate1500plus.com



shane_collinge@yahoo.com

SUBSCRIBE Save money today!

SUBSCRIBE TO LINUX FORMAT MAGAZINE SAVE 50%*

www.magazinesdirect.com/LIN/C75J

Call **0330 333 1113**
and quote **C75J**

NEW!

Digital access to
130+ issues when
you subscribe
to print!**



Save money today! **SUBSCRIBE**



OUTSIDE THE UK?
Turn to page 61 for more great subscriber deals!

» **NEW:** Improved digital back-issue access!**



Get **instant access** on your browser, or Android and Apple iOS devices back to issue 181 (March 2014) with 1,000s of tutorials, interviews, features and reviews.

» CHOOSE YOUR PACKAGE!

Keep your subscription rolling!

Subscribe today and save 50%!* + All-new digital access**

PRINT EDITION



**Only
£21.09**

for 6 months
6 months of *Linux Format*
delivered for £21.09!

DIGITAL EDITION



**Only
£22.71**

for 6 months
6 months of digital
Linux Format for £22.71!

**SAVE!
50%***

Terms and conditions: **Offer closes 15th October 2024.** Please allow up to 6 weeks for the delivery of your first subscription issue (up to 8 weeks overseas). The full subscription rate includes postage and packaging. *Savings are based on the cover price. Payment is non-refundable after the 14-day cancellation period. **Access to the digital library will end with your subscription on print. For a digital subscription, once terminated you only keep the issues you have paid for up to that point. For full terms and conditions, visit www.magazinesdirect.com/terms. For enquiries and overseas rates please call: +44 (0) 330 333 1113. Lines are open Monday-Friday 8:30am-7pm, Saturday 10am-3pm UK time (excluding bank holidays) or email: help@magazinesdirect.com. Calls to 0330 numbers will be charged at no more than a national landline call, and may be included in your phone provider's call bundle.

EXPLORE THE BEST LINUX & OPEN SOURCE SOFTWARE

From the best distros to system security, utilities and fun applications,
we've rounded up everything you could ever need



Ordering is easy. Go online at:

magazinesdirect.com

Or get it from selected supermarkets & newsagents

HighPoint Rocket 1608A

For when **Shane Downing** really needs 56GB/s of storage speed.

SPECS

Bus: PCIe 5.0 x16
Ports: 8x NVMe (eight devices)
Type: M.2 (tool-less)
Form: 2242, 2260, 2280
Cooling: Full-length aluminium heatsink, fan
Power: Yes (PCIe six-pin connector)
Card: Full-height, single-width, 284x110mm
FRU: Yes (VPD data)
Working temp: 0°C–55°C
Power: 82.64W
MTBF: 920,585 hours

If one high-end SSD isn't fast enough for you, how about eight? The HighPoint Rocket 1608A AIC (add-in card) enables you to assemble up to eight PCIe 5.0 SSDs with 16 lanes of upstream bandwidth. With more PCIe 5.0 SSDs and platforms coming to the market as time goes on, there's a natural enthusiast desire to push for more bandwidth, and this AIC delivers. You don't need to use PCIe 5.0 SSDs either; we're using eight PCIe 4.0 Samsung 990 Pros. You could even use a non-PCIe 5.0 slot for that matter.

The Rocket 1608A is an all-in-one solution as it provides cooling, connectivity and an on-board PCIe switch, so you don't have to rely on motherboard bifurcation (see page 62). The card and switch feature everything from indicator LEDs to deeper features, such as synthetic mode. It's quite possible to get 56GB/s or more with the right hardware, and although the price seems steep, it's not unreasonable if you consider the advantage of not needing expensive 8TB drives to reach your capacity goals.

This ideally takes advantage of a full x16 PCIe 5.0 slot and a fast CPU that can keep up. This yields a theoretical maximum bandwidth of 64GB/s in each direction, though real-world performance will tend to be slightly lower.

Rocket powered

The Rocket 1608A uses a standard full-height form factor, clad with a heatsink, fan and eight M.2 slots for NVMe SSDs. To help power the up to eight SSDs, each of which can potentially draw around 12W of peak power, there's a six-pin PEG power connector at the back of the card — the 75W from the x16 slot would otherwise be insufficient. The card and a full drive load can pull up to 82.64W.

The main brain of the Rocket 1608A is the Broadcom PEX89048 PCIe switch, part of the PCIe 5.0 PEX89000 series. The '48' denotes the total number of lanes. Sixteen of these lanes are upstream lanes, from the card to the host system, while the other 32 are downstream. As there are eight M.2 slots available, each one is capable of connecting four lanes at PCIe 5.0.

As noted above, this means it's possible to reach the maximum amount of bandwidth with either four PCIe 5.0 SSDs, such as the Crucial T700 or Crucial T705, or eight PCIe 4.0 SSDs, such as the Samsung 990 Pro.

We used *Iometer* to hammer the HighPoint R1608A with sequential writes for two hours, which shows the size of the write cache and performance after the cache is saturated. One reason you might want such an expensive array is if you really want insanely high sustained performance. These are consumer SSDs, so we're still dealing with pSLC caching. With our eight PCIe 4.0 990 Pros, this does mean it's possible to



Add eight SSD drives for insane read/write speeds but for a cost..

outrun the cache as normal, with sustained pSLC writes up to 45GB/s or so, with TLC dropping to maybe a quarter of this speed. That's not super-amazing but it's certainly possible to get more.

The HighPoint Rocket 1608A is an amazingly fast storage solution when configured with the right drives and the right platform. It's flexible enough to be used less efficiently, depending on your hardware, but it certainly requires a significant investment to reach its full potential. **LXF**

VERDICT

DEVELOPER: HighPoint
WEB: www.highpoint-tech.com
PRICE: £1,580

FEATURES	9/10	EASE OF USE	7/10
PERFORMANCE	9/10	VALUE	5/10

An excellent storage solution if you have the right platform and drives to spare. Impressive, but not for everyone.

» Rating 8/10

Q4OS 5.5

Nate Drake is on cue to explore the latest Q4OS and comes away amazed at its speed and simple setup.

IN BRIEF

Q4OS's nostalgic interface is just the tip of the iceberg. It's a highly customisable OS, perfect for older hardware. The software centre could be slightly more intuitive, though.

SPECS

CPU: 350MHz
Mem: 256MB
HDD: 3GB
Builds: x86-64, i386, ARMHF, ARM64

The Q4OS website describes this distro as a “fast and friendly, desktop-oriented operating system based on Debian Linux”. The current version (code name Aquarius) is based on Debian 12.6 Bookworm. Aquarius is a long-term support release, with security patches and updates until at least June 2028.

The OS can be downloaded as a live DVD ISO as well as a basic install DVD. There's a version with the KDE Plasma desktop but Q4OS is best known for its Trinity desktop environment (TDE). Originally a fork of KDE 3.5, TDE offers a traditional and lightweight user interface.

After firing up the live DVD, we were keen to test this using the bundled *KSysGuard*. On first boot the system used only 180MB of RAM and the CPU load was so minimal it barely registered.

Decisions, decisions

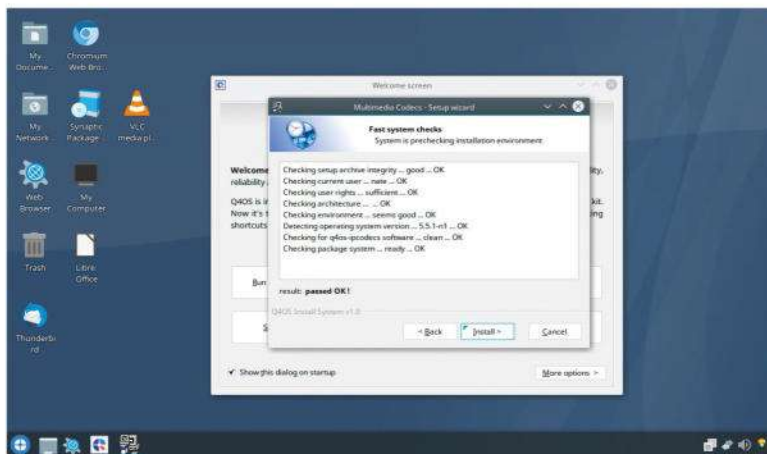
After quitting, we were presented with Q4OS's handy welcome screen. From here you can launch the *Desktop Profiler*, a utility so useful that it has a dedicated manual on the Q4OS website. The profiler's simple interface enables you to choose from a fully-fledged install with web browser, office suite and other common applications, or a basic install with minimal tools, or even a minimalist setup, where you're free to configure your own packages. From here you can also install additional desktop environments including Plasma, Cinnamon, Budgie and Gnome.

Unfortunately, if you're booting from the live DVD, loading profiles via this utility isn't supported, so we opted to run setup instead, which is handled by the *Calamares* installer. Here we discovered that the Software section enables you to choose from a number of install profiles such as Desktop, Live and Pure (minimal). Setup also supports installation of Q4OS alongside other Linux operating systems.

Despite the warning that setup would take a while and the fact that *Calamares* needed to download packages from Debian repos, installation of the Q4OS Desktop version completed in under seven minutes.

On rebooting, we saw that the Q4OS desktop now contained a respectable selection of shortcuts to default apps including *Chromium*, *Thunderbird*, *VLC* and *LibreOffice*. As the OS is Debian-based, we also took the chance to launch the Windows-esque Install Proprietary Codecs wizard from the welcome screen.

From here, you can also click Install Applications to launch the Q4OS Software Center. By default, this



Installing packages such as proprietary media codecs in Q4OS involves using a Windows-style wizard on a case-by-case basis.

offers a small list of 57 featured apps, such as *Okular*, which you can select and install. Doing so launches yet another setup wizard to download the software in question, which seems a little redundant.

You can also launch the *Synaptic* package manager via the *Software Center* to access all other software in Debian's repos. During our tests, *Synaptic* installed the point-and-click classic *Beneath a Steel Sky* in seconds.

This gave us a chance to explore the Trinity menu system, neatly divided into Programs and Documents. Our newly installed game wasn't in the relevant category, but we found it using the built-in search bar.

The built-in *Konqueror* web browser defaults to the Q4OS homepage, where we learned the *Software Center* also offers *Lookswitcher*, which allows for easy switching between desktop themes. The main page also boasts a Dual Desktop option whereby users can install and use both Plasma and TDE. On closer inspection, we discovered this means you can choose different desktop environments from the login window. As useful as this is, it is hardly specific to Q4OS. **LXF**

VERDICT

DEVELOPER: Q4OS development team

WEB: <https://q4os.org>

LICENCE: Mainly GPL

FEATURES	7/10
PERFORMANCE	10/10

EASE OF USE	9/10
DOCUMENTATION	7/10

Offers stability through being based on Debian, a blazingly fast desktop environment and a very easy setup process.

» **Rating 8/10**

Pardus 23.2

Nate Drake offers readers a chunk of Turkish Delight as he tours this Debian-based distro, coded with full government approval.

IN BRIEF

Pardus has stood the test of time and even inspired forks like PiSi. The developers offer carefully curated custom apps and themes. This is definitely suitable as a daily driver.

SPECS

CPU: 2GHz
Mem: 2GB
HDD: 15GB
Builds: x86_64

Pardus began life in 2005 as a fork of Gentoo Linux. Since then, developers based at the Scientific & Technological Research Council of Türkiye (TÜBİTAK) have released successive Corporate and Community editions sporadically.

The latest Community release (23.2) is based on the most recent unstable version of Debian (sid). Spins are available with both Gnome and Xfce. The specs on the left are for the Gnome version.

Downloading the 2.6GB ISO took some time, so we explored the online documentation. Despite being developed in Türkiye, there is an English option, where we learned that Pardus is the Latin word for leopard.

On first boot, the DVD offers you a choice between Turkish and English. You can then boot into the live environment, complete with leonine wallpaper. There is a small number of default apps including *Firefox*, *LibreOffice*, *VLC* and (rather quaintly) *Brasero*.

The desktop also contains a shortcut to launch Pardus's own *YALI* (*Yet Another Linux Installer*). The launch screen offers an Automated Installer but you can also just click Next to choose from one of Pardus's many supported languages. You can also choose to enable LVM and encrypt the system partition.

Coming on strong

YALI forces you to pick a strong password, as we discovered when we tried to set one without 'big letters'. The install process took only four minutes.

Post-install we were greeted by the Pardus Gnome Greeter. This walks you through choosing a desktop style as well as a light or dark theme. You can also choose from 45 wallpapers, and manage icon size.

We should also mention the Extensions section, with add-ons such as a removable drive menu, Caffeine (to disable screensavers) and a Clipboard Manager.

From here you can also launch the *Pardus Software Center*. We noted from the online documentation that while the OS is based on Debian, it has its own repos. Apps are divided into easy-to-find sections such as most popular and Editor's Picks. We used the Software Center to install *MineTest* in under 20 seconds.

After a few minutes of blocky boisterousness, we exited and fired up the *Gnome System Monitor*. We noted that with no other running apps Pardus used around 2GB of RAM, while CPU usage was around 8%.

We obtained the listed system specs from the Pardus site, but anyone wanting to use the Knowledge



The development team has released a special edition of **Pardus 23.2** with a custom theme to celebrate Türkiye's 100th anniversary.

Base must speak Turkish. There is an English version of the forum but the most recent posts are from 2022.

After firing up the OS's native *My Computer* app, we also noted that Pardus's install footprint was just over 7GB. From here you can also view the free space on the root drive, as well as details of any removable disks. The app also supports mounting and formatting drives.

Other Pardus-specific apps are available for install via the *Software Center*. These include *Pardus Power Manager*. The interface allows you to switch between Powersave and Performance, although the Pardus website assures us there are five power modes.

The Pardus category also includes a special theme and custom wallpaper to celebrate Türkiye's 100th anniversary. The main website also offers a special edition of Pardus 23.2 with the theme pre-installed.

Measuring a distribution's popularity can be difficult, which is why we were pleased to see that according to the main website, around 45,000 people have downloaded the Gnome or Xfce version of Pardus 23 so far this year. **LXF**

VERDICT

DEVELOPER: TÜBİTAK ULAKBİM

WEB: www.pardus.org.tr/en

LICENCE: GPL v2 (Community Edition)

FEATURES	8/10	EASE OF USE	8/10
PERFORMANCE	7/10	DOCUMENTATION	5/10

Pardus is a breeze to set up and configure. It's not too taxing on modern systems but has limited online support.

» **Rating 7/10**

Nobara Project 40

Nate Drake delves into this modified Fedora distro. Is this the ultimate in Linux gaming or should you say “no Nobara”?

IN BRIEF

Nobara's definitely tailored to gamers with high-end PCs. Pre-installed packages such as *Wine* and *Steam* make playing a breeze. It's let down by minimal documentation and requiring heavy system resources.

SPECS

LXF suggests:
CPU: 2GHz
Mem: 4GB
HDD: 40GB
Builds: x86-64, ARMHF, AArch64

Creator Thomas Crider, aka GloriousEggroll, describes Nobara Project as “a modified version of Fedora Linux with user-friendly fixes”.

On the main website the former Red Hat software engineer goes on to explain that while Fedora Workstation is a respectable OS, it has little pre-installed third-party or proprietary software. This is a particular issue for gamers, which is why Nobara comes with packages like the *Wine* compatibility layer, third-party codecs for GStreamer and certain Nvidia drivers.

This informs one of the first choices you make when downloading Nobara, as there's both a Standard ISO as well as an Alternative version with Nvidia drivers pre-installed.

The official build of Nobara (the focus of this review), comes with a customised version of KDE. There are also spins using standard KDE, Gnome and two designed for compatibility with the Steam Deck.

The website has an FAQ explaining in detail why network/minimal builds are unavailable to allow users to add their own desktop environment. It also explains that updates are too frequent for torrenting, so ISOs must be downloaded directly from the site.

Beyond this there's no other documentation on the site, which the creators state is a work in progress. However, there is a dedicated Discord channel. Here we discovered that the distro is named after a character in the manga series *Jujutsu Kaisen*.

Unfortunately, neither Discord nor the website had any information on system requirements (the ones listed on the left are for Fedora Workstation).

After booting the 3.9GB Standard ISO, the setup assistant launched automatically, though it took some time to load into Nobara's graphically rich desktop environment. During setup, the splash screen states that the Nobara installer uses *Calamares* rather than Fedora's *Anaconda*. As useful as this information was, most Linux distros take the opportunity during install to state all that is great about that particular OS. You can, however, use Toggle Log to see setup happening in real time. The install process itself took 15 minutes.

Upon rebooting to the desktop, we noted a lengthy loading time for the desktop to appear, complete with a green progress bar. The *Nobara Welcome App* then launched and prompted a system update.

This offers an excellent way to explore Nobara's features. For instance the First Steps section of the *Welcome App* contains shortcuts to launch the *Open*



Nobara is ideal for gamers. It comes with packages not found in Fedora, such as Lutris, Steam and Host Remote Play.

Driver Manager and *Software Manager*. There's even a Web Apps option to convert web pages into apps.

The *Welcome App* recommends installing software such as *Kdenlive*, *Discord* and Crider's own *Proton-GE* for Steam-based play on *Wine*. You can also install new icons and themes, and change the login manager.

Given the system's relatively sluggish performance thus far, we decided to close the *Welcome App* and fire up *System Monitor*. At rest, the OS used around 2GB of RAM and CPU usage was around 65%. The Xfce spin may be more forgiving but this is clearly an OS for high-performance machines. This said, the OS install footprint was small (8GB) compared to Fedora's 40GB.

Special mention should go to Nobara's *Package Manager*, which can be launched via the bottom taskbar. Installed packages are displayed in green. You can also use the search bar to download more, as we did to install *chocolate-doom* and the shareware levels from Fedora's non-free repos. Having criticised Nobara for being a resource hog, it was refreshing to see the install complete in under one minute. **LXF**

VERDICT

DEVELOPER: Thomas Crider et al

WEB: <https://nobaraproject.org>

LICENCE: Free licences + proprietary firmware

FEATURES	8/10
PERFORMANCE	6/10

EASE OF USE	8/10
DOCUMENTATION	2/10

Nobara is graphically stunning. It likely bundles every package a good gamer will need but is also a resource hog.

» **Rating 6/10**

Mabox Linux 24.07

A spin of Manjaro, which is a spin of Arch, is leaving the normally centred **Nate Drake** in a right tizzy.

IN BRIEF

If you're used to a traditional desktop environment, Mabox will present a learning curve. This is balanced out by a comprehensive manual, a handy shortcut list and a slick interface.

SPECS

LXF suggests:

CPU: 2GHz

(dual-core)

Mem: 4GB

HDD: 30GB

Builds: x86_64

Mabox Linux is a Manjaro-based rolling release distro. The developer's About page states that it's based on the latest stable release and uses a rock-solid LTS kernel.

Downloads of the most recent (24.07) and past versions are available via Sourceforge. After downloading the 2.3GB ISO and booting it into a virtual machine, we were pleased to see that not only does Mabox support live mode but you can also choose to run it with proprietary or open source drivers.

These steps are covered in Mabox's user manual, complete with helpful illustrations (<https://maboxlinux.org/users-guide/>). We were less impressed to see that neither the manual nor the extensive forums list the exact system requirements to run Mabox (the specs on the left are for Manjaro).

Box of tricks

As you'll see upon loading the main screen, Mabox doesn't use a traditional desktop environment. The interface is based on the *OpenBox* window manager. It also integrates the *tint2* panel/taskbar and *Jgmenu* for the main menu and side panels.

The main screen also deploys *Conky* to list a series of handy keyboard shortcuts to launch common apps, such as Super+T for *Terminal*. The right-hand *Conky* window slightly mollified us about the lack of clear system requirements in the Mabox documentation, as it displays current CPU and RAM usage. There's also a helpful list of running processes as well as current disk usage. We performed a few routine operations, such as loading websites in the default *Firefox* browser, and saw that RAM usage hovered at only around 1GB.

The top panel contains a shortcut to the *Mabox Linux Control Center*. The main screen provides an easy way to view and edit installed users and packages. There are also options to configure *tint2* settings, such as editing launchers, which we used to add one for *Firefox*. (This is actually redundant as there's a web browser launcher in the top panel.) The *Control Center* also supports configuring both *Conky* and *OpenBox*.

The top panel also includes a launcher for the *Calamares* installer, which we used to install Mabox in a virtual machine. We were pleased to see system encryption is supported. We also noted that you can set a different administrator password during setup. The installation process itself took five minutes.

Upon logging in to our newly installed system, we were treated to the Mabox welcome screen. This



Mabox makes good use of *Conky* to display useful keyboard shortcuts and key system stats such as RAM and CPU usage.

contains links to the user manual and forums. From here you can also click Install Popular Apps, which was welcome given that many of the pre-installed apps aren't found in major Linux distros. This is by no means a criticism, though; *Flameshot*, in particular, is one of the most intuitive screenshot tools we've ever used.

The *Manjaro Application Utility* is compact and well laid out. Programs are grouped into categories such as Browsers and Audio. We dived into the Email category, given there is no default client, and saw that *Evolution*, *Geary*, *KMail*, *Slypheed* and *Thunderbird* are available.

On choosing *Thunderbird*, the installer also offered optional dependencies: the US English spell-checker and support for OTR (Off The Record) messaging. During install, RAM usage remained well below 1GB, so Mabox is definitely more lightweight than Manjaro.

Thanks to using *OpenBox*, browsing through and launching applications is a breeze. You only need to right-click anywhere on the desktop to launch so-called Favorites, such as *Terminal* or the default web browser. The Applications section is neatly categorised to allow for easy launching of apps. **LXF**

VERDICT

DEVELOPER: Mabox Development Team

WEB: <https://maboxlinux.org>

LICENCE: GPL

FEATURES **8/10**

PERFORMANCE **7/10**

EASE OF USE **8/10**

DOCUMENTATION **7/10**

Mabox is easy to install and light on system resources. You may need time to get used to the keyboard shortcuts.

» **Rating 7/10**

Men of War 2

Often found wondering what war is good for, **Jake Tucker** is reminded by *The Management* that it looks after his pay cheque and pension fund.

SPECS

Minimum

OS: Ubuntu

64-bit

CPU: Intel

Core i5 2300

Mem: 8GB

HDD: 25GB

GPU: GeForce

GTX 660

2GB VRAM

Extra: Internet

connection

Recommended

CPU: Intel Core

i7 7700

Mem: 16GB

GPU: GeForce

RTX 3070

The more things change, the more they stay the same. *Men of War 2* layers in new deck-building elements and some shiny graphics alongside a wedge of single-player and multiplayer options, but beyond that, this sequel is the same as the franchise ever was: brutal, shocking violence that can see scores of your forces wiped out in an instant.

This is what *Men of War 2* does best. No other strategy game in this recent RTS resurgence can deliver moments quite like it: a single soldier on a mounted machine gun holding off a hundred enemy infantry; a lucky artillery shell wiping out a terrifying tank; or a brave paratrooper demolishing your opponent's defences by crawling up to a defensive line with a bag of explosives.

These moments leave you feeling elated, and crafting a dynamic sandbox in which they can happen is a huge achievement. Developer Best Way shows a deft eye for creating drama out of what could be meaningless death.

Combat feels mechanically deep and incredibly satisfying when you get it right. Instead of the traditional rock, paper, scissors-style balancing found in many RTS games, *Men of War 2* is even simpler. Bullets kill most infantry without issue, bigger bullets – from an anti-aircraft gun, say, or vehicle-mounted cannon – shred light armour, and explosions kill anything if you hit it right, whether it's a Panther tank or a squad hiding in a barn.

It's your funeral

Fights are large-scale affairs and death is a constant. For every dizzying high, there's a low as your expensive artillery piece is hit by an errant tank shell, or a squad of your best soldiers is caught in the open by an airstrike. These losses could feel meaningless with how often



A road blockade in place.

your valiant guys get turned into jam, but because you're often working with such tight resources, it's harrowing to lose units you know you can't replace. That feeling in your gut teaches you to respect (and fear) the terrifying destructive potential of the game's arsenal.

As battles progress, the beautifully crafted maps stop being just a collection of flora, rocks and buildings, and become a collection of vantage points, kill zones and fatal funnels. Correctly fortified, these points could hold off an entire army, but the chaotic energy means just about anything can happen. Several times we clicked on the map to check out another area on the front, coming back just in time to see an enemy soldier with a can-do attitude has ruined our entire battleplan and reduced our defences to a smoking crater and human detritus. With the shiny new graphics, these body-filled locales look better than ever and it goes a long way to selling the sheer meat grinder that every match turns into.

When everything comes together, it creates thrilling battles of attrition. While the battles in *Men of War 2* feel bigger than ever for the series, a tight cap on how many



A mounted machine gun hiding behind a building.



A US T26 Super Pershing supports the more common M4 Sherman.

units you can call in (and the resources needed to call things in) means you constantly feel starved of troops. After a particularly bad loss in an area, you often find yourself unable to bring in reinforcements, turning everything into a hopeless scramble for survival as you send a few scattered and ill-equipped infantry to try to tackle whatever godless killing machine has just broken through your front line.

New units are slowly unlocked. You are first able to call in the First Echelon soldiers, then you can call in a greater number of more equipped Second and Third Echelon troops, these Echelons effectively gating your higher-powered units behind a timer. These moments create a small surge as you can finally call in a big tank or an artillery piece that you think could change the flow of the battle. Generally, though, we found most battles become much less intense after first combat. Most matches end not with a bang but with a whimper, as your dribble of surviving troops desperately tries to advance.

Similarly, the story-led campaign gives you so little to work with resource-wise that every death feels like something you should quickload to get around. An enemy sniper can take out several infantry before you can properly react, and when you're playing with a small fighting force, it feels like you're often setting yourself up for failure. We've played several missions in the story-driven campaigns and sampled a bit of the Russian, German and US missions, as each has a unique storyline, but frankly we found that they just irritated us. The set pieces were impressive, but with so little room for error, there's not a lot of room for experimentation.

This was less of an issue in other campaigns – which include dynamic and procedurally generated campaigns – but there was always a nagging feeling that there just weren't enough troops to do what we wanted. A skill issue, yes, but we've put hundreds of hours into *Men of War: Assault Squad* and never felt this in previous games.

The big new addition to *Men of War 2* is the battalion system. This enables you to bring multiple battalions with different selections of units into the game's two dynamic campaigns and to the wealth of multiplayer matches on offer. These battalions are pretty self-explanatory: an artillery battalion has a lot of artillery, while players selecting a tank-led option will no doubt be unsurprised to find themselves knee-deep in tanks.

Battalions can also be edited and have rules not unlike the faction system of Eugen's *Wargame* series, so your infantry deck can have a lot of diverse options and your tank battalion has, at the risk of over-egging one joke, several tons of tank to choose from. To do so, you need to get currency – by completing achievements or matches – and then sub units out one by one. It's byzantine, and while it will no doubt be appreciated by the sort of people who play games like this for years, we'd prefer the idea of having some boundaries, and found this deck building overwhelmed us with choice.

You can see this in the two procedurally generated campaigns, too. Conquest enables you to take several



battalions on a romp across Europe, with a battle each turn. *Raid* is a smaller-scale affair generating 16 maps to fight through. *Raid* is billed as more accessible, but because both require you to research new technology, then build a deck over time, it still feels very involved. Still, these dynamic options should be more replayable than the narrative campaign, and give players who do want to tweak their battalions a perfect toy box.

Despite these missteps, it's hard not to enjoy what *Men of War 2* is putting down. While *World War 2* has proven to be fertile ground for the RTS genre, *Best Way* has created an uncompromising and compelling RTS. Sadly, many of the new additions to the *Men of War* formula don't feel particularly necessary. As bizarre as it feels to say, we find ourselves wishing that *Best Way* had tried to reinvent the wheel less, and built more upon the brutal combat that underpins the game.

Instead, *Men of War 2* is an uneven package. There's no denying the wealth of stuff here: the game is full to bursting for RTS fans willing to learn the razor-wire-covered ropes. We had a lot of fun, and despite how punishing it was, we found ourselves eager to hop in again and again. We wanted to unlock all the different battalions and see how they feel in combat, and we're even tempted to spend more time in the game's multiplayer, despite knowing we'll get battered.

Not that losing would be a surprise. Failure comes in many forms in *Men of War 2*, and failure can be fun. Sadly, several small flaws combine to make *Men of War 2* difficult to love, despite the tight mechanics that hold everything together. **LXF**

The sheer number of supported armament types is astounding.

VERDICT

DEVELOPER: Best Way

WEB: www.fulcrumpublishing.com

PRICE: £38

GRAPHICS 7/10

GAMEPLAY 7/10

LONGEVITY 8/10

VALUE 7/10

A flawed but rewarding gem for masochistic RTS fans.

» **Rating 7/10**

Roundup

Chrome Remote Desktop » X2Go »
Moonlight » Remmina » RustDesk



Michael Reed

is a technology writer, programmer and multi-faceted Linux fiddler, fiddler and facilitator.

Remote desktops

Ever wish you had the desktop of another computer right in front of you? **Michael Reed** checks out five applications that can do this and more.

HOW WE TESTED...

In each case, we are primarily interested in the client applications, but in some cases, the client is tied to a specific server, which we'll make some mention of.

Installation was the first thing we looked at. As a minimum, we like to see a simple way of installing the latest stable version of an app, and that installation should be documented on the website. We think that's a fair ask on our part.

In each case, we tested the remote desktop client with the appropriate server, whether that was an official server that it was intended to be used with or a generic server that it is able to work with. We tried wired and wireless connections on our local network, and we also tested making the connections across the internet.

We tried to use the systems and tried to get an idea of typical performance under a range of situations. We added remote connections and considered what the day-to-day workflow would actually be like.



Remote desktop solutions enable you to control one computer from another. They do this by running a server on the remote computer that compresses the graphics and sound, and sends it over the network. On your end, you run a client application that views the remote desktop and sends your keyboard and mouse input back to the server. In other words, you are in control of the other computer and can operate it as though it were the computer in front of you.

You can use a setup like this for remote administration, technical support or even

gaming. You might be installing printer drivers for your mum or gaining access to a more powerful computer from your laptop or mobile device.

Some of the solutions can connect to a generic server through open standards and some of them are tied to a specific server, but we're primarily focused on the client applications this time around.

We've got five solutions for you to consider. They are all useful, but they have a range of features and approaches, and the best solution for you will probably be the one that's closest to your needs.

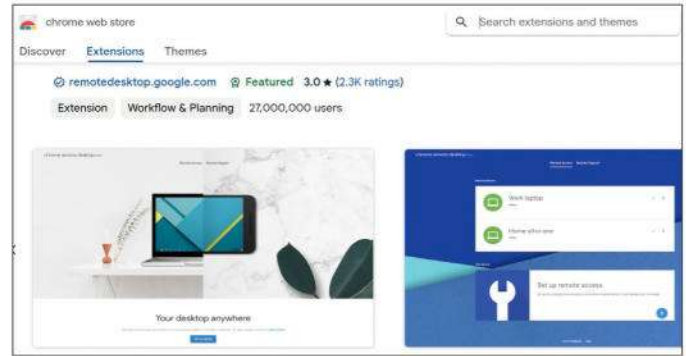
Installation options

Ease of installation and as many supported platforms as possible.

Moonlight has an extensive set of supported platforms, reflecting its gaming origins. It can be installed on all the major desktop platforms, and there are homebrew ports for the major consoles and some handhelds, along with single-board computers such as the Pi. For our Ubuntu machine, we installed it via Flatpak but the Snap was up to date, too.

We installed X2Go via a PPA. There is more than one client, but we followed the instructions on the PPA homepage and installed the official X2Go client, which is a QT5 app. The client machines must have SSH access to the server computer, which you might have to set up. You also have to install a desktop environment that X2Go can launch on the server, and Gnome 3 doesn't work properly. So, we installed Xfce on top of Ubuntu's Gnome 3, and we had to switch off compositing to make it work. With extra work, it can be installed on Mac OS and Windows.

The Remmina website details the status of builds in the repository for dozens of distros, but it also includes instructions for making a Flatpak installation, which we did as the Linux Mint version was slightly out of date. There are also instructions for Snap, but that involves manual configuration at the command line to work around the sandboxing. It's a Linux-only program.



To install the Chrome Remote Desktop client, one has to install the extension within Chrome. On the server side, a DEB must also be installed.

To install *Google Remote Desktop*, we installed the *Chrome* browser extension on the host and remote machine. If you can't use *Chrome*, *Google Remote Desktop* also has a desktop app that runs its own window. The server is more complicated to install on non-Debian-based distributions.

The simplest way of installing *RustDesk* is with the *Applmage* on both sides of the connection. This uses the *RustDesk* server, but it's also possible to self-host. There are other options such as *DEB* and *Flatpak* and it can be installed on *Windows* and *Mac OS*.

VERDICT

CHROME REMOTE DESKTOP	6/10	REMMINA	7/10
X2GO	8/10	RUSTDESK	8/10
MOONLIGHT	10/10		

The fully open source options offered the broadest installation options.

Compatible servers

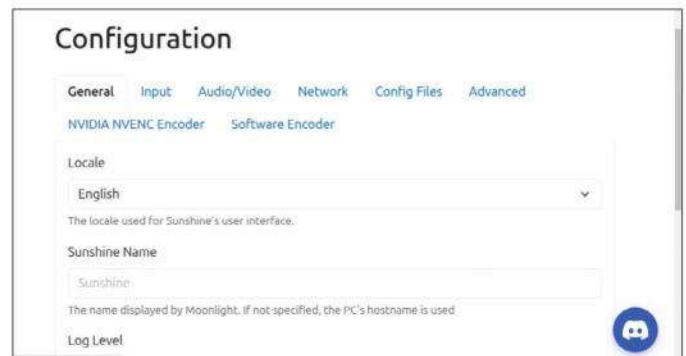
Every client needs a server to connect to in order to be useful.

Remmina doesn't have its own server, but it has support for the main protocols that you're going to encounter: VNC, RDP and SSH, and there are a few different Linux server options for each of these. It can even be used as a client for X2Go.

Moonlight is designed for the proprietary Nvidia GameStream protocol, but it is also compatible with *Sunshine*, an open source server that implements that protocol. Even though it only supports the one protocol, this means that paired with *Sunshine*, it offers an end-to-end open source solution for desktop and game sharing.

X2Go uses a modified (and incompatible) version of the high-performance NX protocol that is implemented by the semi-proprietary *NoMachine* remote access system. It provides its own server.

Google Remote Desktop uses its own, proprietary, protocol. The user isn't required to do any server configuration. A Google login is required on the server and client, but they can be the same Google account if you own both machines. Google locates the computers on your LAN or across the internet.



Sunshine is the other half of the *Moonlight* solution, if you want to stay fully open source for the client and the server. It uses a web interface for configuration.

RustDesk uses a custom protocol. In actual use, what we would normally think of as the server and the client are the same program, and *RustDesk*'s ID server allows the two to find each other. However, *RustDesk* encourages you to set up your own server and this is more appropriate if you want to use *RustDesk* over the internet. We rate setting up your own server as a slightly complicated affair. Paid server options are available.

VERDICT

CHROME REMOTE DESKTOP	6/10	REMMINA	8/10
X2GO	7/10	RUSTDESK	8/10
MOONLIGHT	7/10		

Remmina connects to most of the popular protocols. The other options have their own servers.

Session management

The user's first port of call.

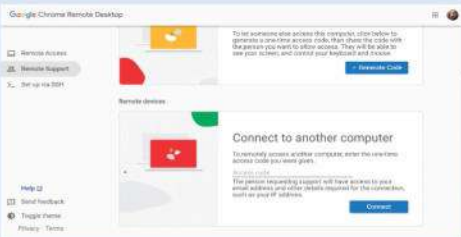
Session management is the first point of contact for daily use of a remote desktop tool. It's the part of the program that lets you set up a connection to another computer and manage a list of such connections. What you are looking for here probably depends on what you are using a remote desktop for.

The home user might only have a couple of different machines to which they need quick access. These might be a desktop PC and a couple of virtual machines, for example. We could extend that to the machine of a relative for whom they provide technical support. A business or IT expert user, on the other hand, might be more interested in good facilities for connecting to a large variety of computers.

In all cases, a variety of configuration options enables you to tailor the bandwidth usage for different network types, such as a hardwired LAN, a Wi-Fi connection or access over the internet.

Google Remote Desktop 6/10

Google Remote Desktop is designed for single connections to a remote computer. It's a good system for technical support as the user can send you a code number that immediately allows you to open their desktop on your browser. Although there isn't much that you can change about how the connection works, there are many situations where options such as 'share this desktop' and 'connect to a computer' within a browser will be a welcome sight. It's not designed to log into a large collection of other computers without the cooperation of the users. The user whom you are supporting conveys a password via a route such as email or over the phone. The client who would like access to that computer uses the password, and Google services figure out where the two computers are located on the internet. The available settings are far from a tweaker's paradise, but some people prefer a simple solution, in all fairness.



X2Go

7/10

X2Go doesn't work quite like the other apps because it creates a session on the server computer and this uses a unique environment with its own filesystem. So, you've got a blank desktop that uses the desktop environment of your choice. Files on the remote persist between sessions. Creating a new X2Go session is easy: click on the appropriate icon on the toolbar. This pops up a dialog with tabs for the different sections. The information on the panels is more focused on the details of the session, such as the desktop environment, than is typical of a program of this type, but that's to be expected because of how X2Go works. Handily, there is always an option to create a desktop icon of the session. The options are slightly slimmed down compared to something like Remmina, but they are probably detailed enough for a typical variety of network speeds, and they are well presented and easy to use.



File transfers and cut and paste

Cut and paste between client and remote is invaluable.

A Moonlight/Sunshine combination doesn't come with any file transfer facilities built into the user interface. It also lacks a cut-and-paste mechanism that can work over the network. If you really needed this functionality, you could set it up yourself (via SSH for example).

The file transfer facilities of RustDesk are built into the client app. To transfer from the remote, the user selects from the overhead icon menu that is part of the connection window to open a two-panel file manager showing files on the client and remote. Neat. There is also a keyboard combination for remote cut and paste.

X2Go takes a different approach, because a shared folder is set up in the session preferences. You can create a shared folder while the session is running, but it's not mounted until the session is restarted. Once it has been created, a new folder appears on the backdrop and this is linked to the folder that you've selected on your client machine. It has options to share the clipboard.

Remmina works in much the same way, offering a shared folder that can be used to transfer files. It also has a function to send the current clipboard item as though it was typed into the remote computer.

Chrome Remote Desktop doesn't have a file transfer utility. Arguably, this isn't a massive handicap because you can't use it without a Google account, so you could use Google Drive to transfer files between computers. It's a lot of extra mouse clicks compared to a well-integrated file transfer tool, though. However, it does have a clipboard synchronisation system.

VERDICT			
CHROME REMOTE DESKTOP	5/10	REMINNA	7/10
X2GO	8/10	RUSTDESK	8/10
MOONLIGHT	3/10		
Moonlight lets itself down in this area, but the others offer good facilities through different methods.			

Moonlight**7/10**

When *Moonlight* is first launched, you are presented with an icon-driven selection of servers that lead to another set of icons relating to the session type you wish to launch on that server. The way this breaks down is that these sessions can be a simple takeover of a running desktop, or a command can be run first to load a program such as a game or an emulator.

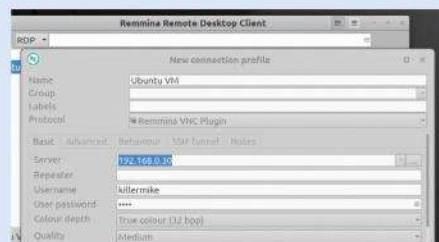
Rather than accessing the session settings from the client, they are configured on the server side. This is done so that the connections are consistent between different devices and clients, which makes sense given the intended use of this system. *Moonlight* includes a useful variety of bandwidth options.

It's a powerful system for most home users, which seems to be the intended use scenario of this software, but we're not sure how well a system like this one would scale for an office or technical support role with a lot of machines.

**Remmina****10/10**

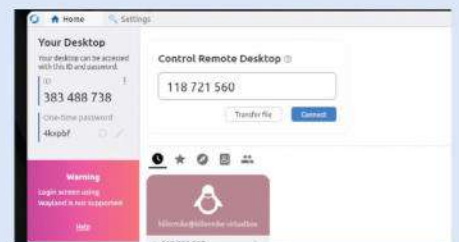
Along the top of the *Remmina* window is a quick launch bar for making one-off connections. Most of the time, it's better to permanently add a server by clicking on the + icon. Once you have connections stored in *Remmina*, they are listed in the main window and are searchable, and they can be sorted into groups, too.

When it comes to configuration of each session, we'd classify the available options as deep, meaning that you can tweak every aspect of the connection. The way it works is that when you select a different protocol from the drop-down menu at the top of the configuration dialog, the options underneath also change. These tweaking options enable you to adapt to connections of varying quality and speed. Beyond the fine-tuning, the wide selection of options means that this tool can be fitted into any existing setup, in cases where you're not in complete control of the remote machines.

**RustDesk****7/10**

We like the aesthetic of the *RustDesk* client, and it's fairly apparent how the session interface works when you first encounter it. You can quickly connect to another machine by typing in the ID number displayed by the remote machine. When first launched, the interface is in icon mode, and two types of list mode are available. It all adds up to a pleasant-looking interface that will appeal to non-experts with a few connections or business users who need access to a large collection of remote computers.

As it's part of a closed system of server and client, it's not surprising that it doesn't have a huge number of options for each session and relies more heavily on global options for the program itself. Some more technically-orientated users might miss *Remmina*'s fine control, but the essentials for quality and performance balance, to match your overall setup and network connection, are included.



Documentation and community

Getting the software set up and getting help when you need it.

The *Remmina* website links to an official manual, but there's not much to it and it hasn't been updated since 2018. The site has a link to a Matrix-based chatroom, but the link didn't work. *Remmina* also has its own Reddit subreddit, but it doesn't have a lot of traffic and questions go unanswered. The saving grace in terms of documentation is the wiki, which has 57 pages that cover most areas of the program.

The X2Go website also features a wiki that covers the basics of operating the software. A Google search found us a few tutorials on installing and using the software from various sources. The user and developer community arrange both virtual and real-life meetups, so it's certainly a project with a community around it.

Maybe it's because it's used by gamers, but *Moonlight* has an extremely active Discord community. The wiki covers all areas of the program. The subreddit is active, with most questions being answered and plenty of ongoing discussions. It's obvious that it's going to be easy to get answers to support questions.

Chrome Remote Desktop doesn't have much of a community around it. Generally, if you run into problems with the software, searching on Google will take you to the appropriate page on Google's Help centre.

RustDesk has a healthy overall documentation and support scene. The website documentation covers topics such as self-hosting and various enterprise issues, such as setting access control. The Reddit subreddit has quite a lot of traffic in it, as does the Discord server. There is also paid support for those who want it.

VERDICT

CHROME REMOTE DESKTOP	5/10	REMINNA	5/10
X2GO	6/10	RUSTDESK	8/10
MOONLIGHT	8/10		

All of the options cover the basics, but *RustDesk* and *Moonlight* have the most active communities.

Mobile control

Connecting to a remote desktop via a phone or a tablet is damn handy.

Remmina doesn't have a mobile app version. As it is a client that supports a number of open protocols, you could install another remote access app on your mobile device to take over when you're on the move.

Moonlight is available as a mobile app. The app has gaming-specific features such as being able to add an on-screen game controller. However, it's quite possible to use the app to control a full desktop, and therefore serious applications, particularly for light or occasional use in this role.

Google provides a client app to access a *Chrome Remote Desktop* shared desktop from a mobile device, with Android and iOS versions. Its reputation has suffered recently as updates have removed functionality and compromised its stability.

There is a *RustDesk* Android app, but at the current time, it is unavailable in Google Play Store, apparently due to scammers persuading people to install the app. Check the official download page for updates on this matter. On the other hand, the *RustDesk* app is featured on the Apple Store for iOS.

X2Go makes extensive use of X11 to provide remote virtual desktops, and it's not that surprising that there isn't an Android



The Remmina app has much the same features as the desktop clients. The different sessions such as Desktop and Steam are set up on the server.

or iOS application. There is talk on some of the forums that it can be done by installing a Linux environment on your phone, but it's a complicated procedure. Basically, our advice is to choose one of the other options if you need access from your mobile device.

VERDICT			
CHROME REMOTE DESKTOP	6/10	REMMINA	4/10
X2GO	3/10	RUSTDESK	5/10
MOONLIGHT	8/10		

Moonlight has a good app, particularly for gamers. The others are either MIA or of dubious quality.

User interface and experience

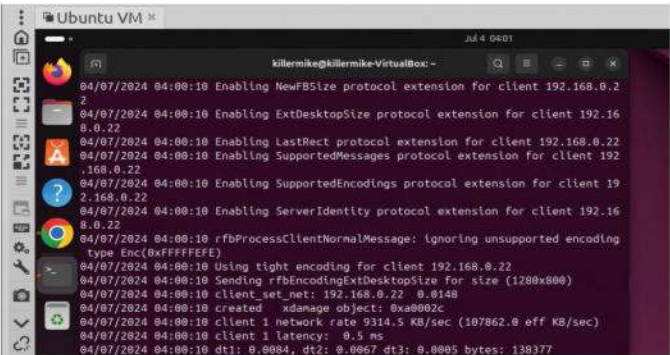
What is it like in actual use?

The main *Remmina* window has icons down the left that link to useful functions such as changing the connection quality on the fly, resizing and changing the scaling of the viewer, and sending the clipboard to the remote machine. It's also a multi-tabbed window, which can be extremely useful.

RustDesk has a little icon bar at the top that expands in use to offer quite a lot of features, such as changing the screen settings in real time and opening a chat window. It's a completely usable free application, and it feels thoroughly professional in use.

The main *Moonlight* window doesn't have a control toolbar and is controlled through keyboard combinations. Before starting to use it, learn a few key commands so you can move between full-screen and windowed mode, if nothing else. If you want to change things like the video compression quality, you have to quit the connection and go back to the client configuration. *Moonlight* isn't designed for system admin purposes, as such, although, it can be used in that role if the requirements are basic. For gaming on a LAN, you probably want to ramp up the bandwidth options.

Much the same can be said for *X2Go* as there are no real-time controls on the main window, although there are suspend and quit controls on the session launcher window. There is also a log



Remmina has a set of controls accessible through an icon bar on the left. They are extremely handy for accessing functions without having to learn keyboard combos.

window to aid with troubleshooting. The standard window doesn't take over the mouse by default, so new users won't end up stuck without seeing how to get control back.

The *Google Remote Desktop* workflow is simple and takes place within the web browser. Once the connection is up and running, there are no on-screen controls, but you have the remote desktop available on a browser tab, a convenient arrangement.

VERDICT			
CHROME REMOTE DESKTOP	5/10	REMMINA	8/10
X2GO	6/10	RUSTDESK	8/10
MOONLIGHT	5/10		

Remmina and RustDesk have on screen controls, a bit like some virtual machines, which are intuitive to use.

The verdict

Remote desktops

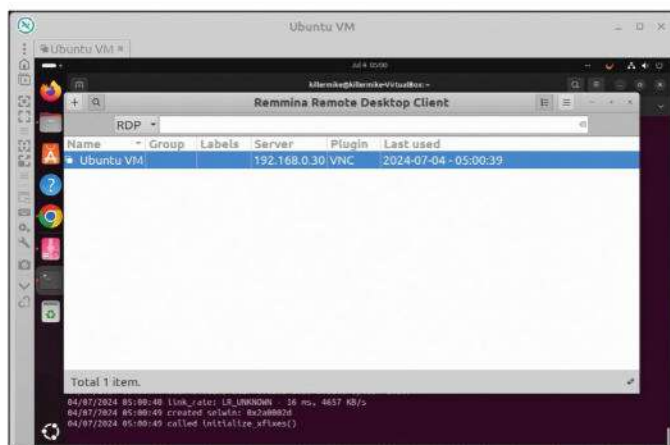
We've ended up with five options that can handle a variety of remote desktop scenarios. *Remmina* is best if you need to connect via a variety of protocols. Having said that, there's nothing about the interface that's too complicated for casual users who need to occasionally connect to another computer. The detailed configuration options are there to fine-tune how things work on different types of connection, but it's up to you to set up a server on the remote side.

Moonlight started life as a client for Nvidia's GameStream protocol, and thanks to the Sunlight server, it's a complete open source system for establishing a remote desktop. It has a lot of useful options to fine-tune the connection, which makes it adaptable for different usage scenarios and types of network. It also has gamer-specific features and the advantage that the client is available on a wide variety of hardware platforms. It's our first choice as an all-rounder for typical home users, including gamers.

RustDesk is an easy-to-use system, and the setup can be as simple as copying the software to either end of the connection and running it. It's completely open source, and you're encouraged to self-host the ID server for quicker connection and enhanced privacy. There are enough options to configure *RustDesk* to best balance performance and quality.

X2Go works a bit differently from the other options that we've looked at in that it creates a Linux desktop that is served, over the network, to the client application. It's a handy tool for creating remote desktops on the fly. Making things all the more impressive, it can even work in the same way as a traditional remote desktop tool and take over a running desktop. It's a shame that it's inherently difficult to make a mobile version of the client.

If you're using *Chrome* as your web browser, setting up a *Google Remote Desktop* client is as simple as adding an extension to the browser. We had mixed results, at best, trying to work *Firefox* into this system. There's a bit more to it on the server side as you have to install a package on that system. Once up and running, it is a simple system that's well suited to carrying out remote tech support, but it doesn't offer much in the way of configuration and fine-tuning. It's simple but not very flexible.



1st **Remmina**

9/10

Web: <https://remmina.org>

Licence: GPL-2.0-or-later **Version:** 1.4.35

Works with most standards. Lots of options to balance performance.

2nd **Moonlight**

8/10

Web: <https://moonlight-stream.org>

Licence: GPL-3.0 **Version:** 6.0.1

Good for home users. Gaming features. Open source client and server.

3rd **RustDesk**

8/10

Web: <https://rustdesk.com>

Licence: AGPL-3.0 **Version:** 1.2.6

Simple setup. Optional commercial hosting and support. Chat feature.

4th **X2Go**

8/10

Web: <https://wiki.x2go.org>

Licence: GPL-2.0-or-later **Version:** 4.1.2.4

Quick setup of empty custom desktops. Can connect to existing desktops.

5th **Chrome Remote Desktop**

7/10

Web: <https://remotedesktop.google.com>

Licence: Proprietary **Version:** Current web version

Simple solution for technical support. Works in browser.

» ALSO CONSIDER

If all you need is command-line access to another computer, installing an SSH server might be a better option than a remote desktop solution. The advantages are low CPU usage, practically no latency and perfect native text quality. Look at the instructions in the documentation for your chosen distribution to figure out how to set up SSH.

Some of the virtualisers have built-in facilities to provide a remote desktop experience. For example, *VirtualBox* has its

own RDP server, meaning you can make the desktop available to other computers.

Some of the commercial systems, such as *TeamViewer* or *RealVNC*, offer a free tier that might do what you need.

It's a shame we couldn't fit *Apache Guacamole* into the line-up as it's an interesting-looking remote desktop system that can use protocols and HTML5, meaning you can use a web browser to view and control the remote machine. **LXF**

Fresh Mint 22!



Linux Mint is in season and **Jonni Bidwell** has been sniffing it voraciously. Join him on a fragrant voyage of discovery.

We've been nothing but consistent in our recommending of Linux Mint, to beginners and old hands alike. Linux newbies will find it easy to learn, and power users of other operating systems will enjoy the control and configurability that it offers. As the project enters its eighteenth year, it continues to go from strength to strength. While the world of desktop Linux has grown out of its wild years, where each new release would bring exciting (often scary) new features and

changes, there's still a bunch *[hmm – ed]* of stuff to get excited about with Mint's latest outing.

Built on the rock-solid foundation of Ubuntu 24.04, Mint offers a fresh new perspective on what a Linux distribution can be. It's fast, it's free and (unlike certain OSes from Redmond we could mention) it's privacy-conscious. The Cinnamon desktop environment is better than ever, but if you prefer MATE or Xfce, those editions are also better than ever. Thanks to an overhauled *Software Manager*, you get access to not just the

Ubuntu software library, but also a whole plethora of verified applications from the Flathub platform.

We'll walk you through booting, installing and making the most of this delectable operating system. We'll cover setting up *Timeshift*, so you can easily undo any system administration errors you might commit. We'll introduce the gentle art of managing everything from system updates to sound servers to touchscreens. And we'll show you how to get help if things go south. The best part about Mint is the community.

CREDIT: Magictorch

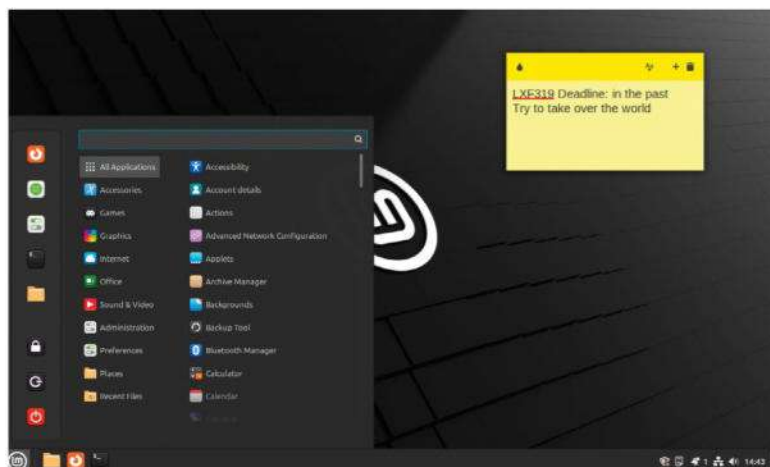
Mint is for its users

Discover why Linux Mint is your new favourite Linux flavour.

New Ubuntu Long Term Support (LTS) releases are a particular pleasure. Once the thrill of the launch has passed, we can bathe in the afterglow as a wave of new Ubuntu derivatives (such as Pop!_OS, Elementary OS and KDE Neon) washes over us. Most refreshing among these wavelets must surely be Linux Mint. This community-powered distribution has stuck to its ideal (giving users what they want; radical, we know) since its genesis in 2006.

If you're new to Linux, Mint 22, code name Wilma, is an ideal place to start. It's easy for beginners to learn and its Cinnamon desktop makes it simple to pick up if you're used to Windows or Mac OS. If you like, but don't love, Ubuntu, Mint might be for you. It uses Ubuntu's low-level packages to power the system, but does away with the Gnome desktop and Snap-centric package manager. If you prefer a slightly more lean desktop environment, Mint provides Xfce and MATE editions, too. There's also Linux Mint Debian Edition (LMDE), but we covered that in **LXF313**, so check that out if you'd like to run Mint on 32-bit hardware.

Mint's job has in ways become harder as Ubuntu has evolved. As low-level plumbing (package management and audio) become ever more inveigled with the desktop, getting that plumbing to work elsewhere becomes increasingly onerous. With Mint 22, though, those efforts have paid dividends. The team has wired



up the PipeWire sound system (ousting PulseAudio), incorporated the libinput touchpad driver (deprecating Synaptic) and banished Ubuntu's Snap Store (in favour of its own *Software Manager*). The devs have also been working on XApps, their desktop-independent core app suite (including but not limited to a text editor, image viewer, document reader and video player).

Being based on Ubuntu comes with its advantages though. It's widely used and extensively tested. There's the five years of support, so if you like Mint, you won't have to upgrade your operating system until then. Bugs and security issues are patched swiftly, and you're gently reminded to apply these fixes automatically. Unlike Windows, that update process is entirely under your control and won't hijack your computer for hours.

There's a huge collection of Ubuntu software in the traditional Apt repositories, plus you'll find even more as Flatpaks (the modern, distro-independent packaging system) in Mint's *Software Centre*. We cover installation over the page, but first grab a USB stick and download the Mint 22 ISO image from www.linuxmint.com/download.php. Then see the box (below) to combine them into a bootable installation medium.

Mint's Cinnamon desktop will have you feeling right at home, including if at home includes yellow notes.



Balena Etcher is a simple way of making bootable Linux media.

» ETCHING A LINUX MINT USB

Depending on your OS, there's any number of programs you can use to write the Mint ISO file to a USB stick. We'll cover *Etcher* because it's cross-platform and gets the job done. Other tools (such as the more advanced *Rufus* on

Windows, or your Linux distro's USB utility) probably work, too. You need at least a 4GB USB stick as the image is around 2.9GB. Plug it in and don't worry about formatting it (the ISO file is its own filesystem). Make sure there's

nothing important on it as it's about to be wiped.

Download *Etcher* from <https://etcher.io> and fire it up. Point it at the **linux-mint-22-cinnamon.iso** file you downloaded, then select your USB drive – the installer helpfully hides

any non-removable storage, so you don't accidentally wipe your main drive. Hit Flash and watch the progress. Or go and fetch a cup of tea. It'll take a couple of minutes on a USB 3 device, or longer for an older medium.



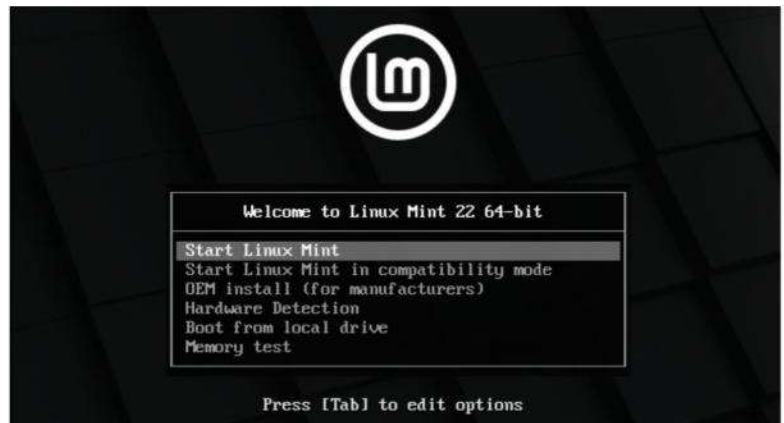
How to install Mint

Get Mint booted, explore the live environment, then get this herbaceous powerhouse installed forthwith.



By this point, you're more than likely eager to try out Mint for yourself. You might want to do that in a virtual machine (there are some pointers for running in *VirtualBox* at www.linuxmint.com/rel_wilma.php). But you might also want to run it directly, which is easier and totally risk-free. The Mint installation image boots to a live environment that doesn't touch your storage until you elect to install it. In order to get there, you need to figure out how to get your machine to boot the installation USB we created on the previous page (yes, turn back and do that now – you know who you are).

This usually involves pressing a key (often F10, F2 or Delete) as the machine turns on, in order to either summon a boot menu or enter the BIOS/UEFI settings and change the boot order manually. The process is extremely motherboard-dependent, but also some UEFI setups jump straight into the OS's bootloader without giving you a chance to change boot settings. On Windows, you can get around this by holding Shift as you push the shutdown button and selecting Troubleshoot > Advanced Options > Firmware Setup. If you're using Linux, there might already be a UEFI Firmware Settings option at the bottom of your boot



Hopefully, after a brief battle with firmware settings, your USB boots and you see this screen.

menu. Or you can force the issue by running:

```
$ sudo systemctl reboot --firmware-setup
```

Assuming you figure that out, you'll see the Mint boot menu, from which you should select Start Mint (unless you have an Nvidia graphics card, in which case check the box – below). The boot process should spring into life and you should eventually be greeted with a chime and the monochromatic stylings of the live environment. This behaves almost exactly like a real install of Linux Mint, albeit a tad slower and with some space limitations (the virtual filesystem lives in 2GB of system memory). We perhaps should have noted earlier that Linux Mint requires a 64-bit processor and recommends at least 4GB memory for comfortable usage. It will work with less, but will not be fun. Most machines from the last 15 years will happily run Mint, but we wager some of our readers have much older hardware. There are more slimline distributions that are much more suitable for these devices.

Clicking the Mint logo in the bottom-left brings up the applications menu where, if you rummage around, you'll find such gems as *Firefox*, *Thunderbird* and the *LibreOffice* suite. Plus there's a raft of other tools and utilities to explore and do things with your system. The Cinnamon desktop is extremely customisable, so it's worth exploring the Applets, Desklets and Extensions available from the Preferences menu. You can also change the desktop cosmetics from Preferences > Themes, including changing the Style (between the older Mint-X, Mint-L, formerly Mint-Y-Legacy, and the shiny new Mint-Y) and colourings.

If you need to change your resolution, or otherwise tweak things to do with your monitor(s), right-click the desktop and select Display Settings. For other settings, use the *Control Center*, which is the third icon from the top in the application menu, also available from Preferences > System Settings. We'll cover these post-

» NVIDIA DEVICES

Some users of Nvidia graphics hardware may have difficulty booting Linux Mint. Plenty of them will have no issues (and shouldn't worry about these steps – the open source Nouveau driver will take care of you), but some new or very old cards may boot to an unhelpful black screen. Should this happen, some simple remedial action might help.

Hit E at the boot menu (with Linux Mint highlighted) and add `nomodeset` to the end of the line beginning `linux /boot/vmlinuz...` towards the end of the configuration. This boots the system without kernel modesetting, which may make the desktop unstable. But we only need a couple of minutes to go to Administration > Driver Manager, install Nvidia's proprietary drivers, and then reboot. Then no more `nomodeset`. If you have a hybrid (Optimus) graphics setup, you should find that works now, too, and can be controlled from the system tray.

If that doesn't work, you'll want to revert your changes in *Driver Manager*. The release notes (currently) suggest trying `nouveau.noaccel` instead of `nomodeset` or even adding the more extreme `noapic noacpi nosplash irqpoll` instead of `quiet splash`, which disables power management and IRQ polling.

install, but it's a good idea to test as much of your hardware as you can to avoid any annoyances later on. Don't worry too much about poor graphics performance (unless it's so poor that things break); you can find Desktop Effects in the Preferences menu and disable them to lessen the burden. It's very likely a different graphics driver will be installed, so hopefully any issues will go away after a reboot (or two).

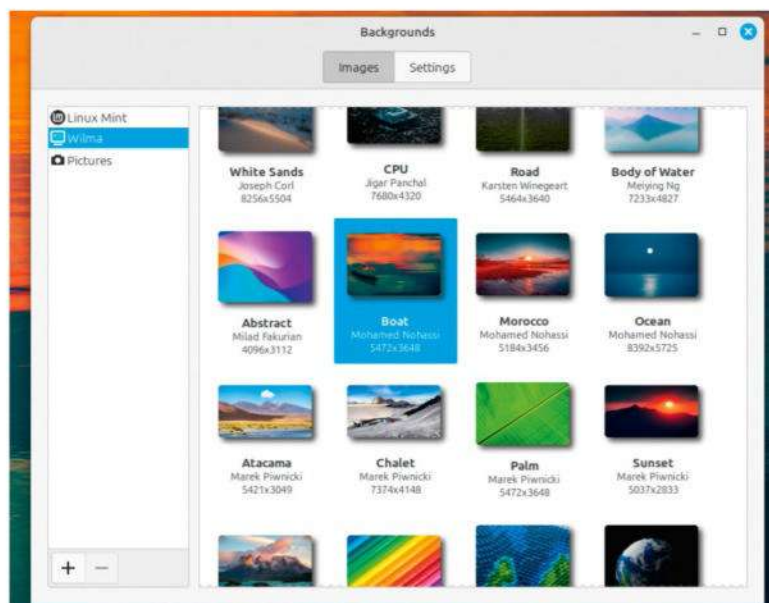
The *Files* app (aka *Nemo*) can show you any NTFS volumes (if you are installing alongside Windows), or any other kind of storage you might use, including network shares. So, it's a good idea to have a play around and see which of your files (particularly documents and multimedia) work well in Mint. Again, don't worry too much if something seems awry. Further multimedia codecs will be installed later if you tick a box. Most NAS boxes support CIFS sharing, so should be visible in the Network section. Oh, and don't forget to check out all the glorious new wallpapers by right-clicking the desktop, selecting Change Desktop Background and navigating to the Wilma section.

If you want to chat in real time with real-life Mint users, go to Internet > Matrix. This *Matrix* is a web app (which uses the open source Element platform and Matrix messaging protocols) and the de facto home of the old #linuxmint IRC channel. You need to create an account to use it, but this only takes a couple of clicks. When you're done, you're invited to join the Linux Mint space, where you can read and chat about all things Mint. Discussion is separated into different Rooms for different projects or levels of seriousness. Why not start by introducing yourself in the Introductions room?

Installing Mint

It's possible to install Mint alongside Windows, but this is not risk-free. If you're new to installing Linux, we recommend doing so on a different machine from one on which you do actual work. Failing that, if you must install on to a machine you rely on, get a separate drive for Mint. A £25 240GB SSD will be just fine and plenty fast. By avoiding any repartitioning on your main drive, the risks of making Windows cry are greatly reduced. But make sure to back up anything of value. Just in case. That advice administered, let's get this show on the road by double-clicking the Install icon on the desktop.

It's worth taking up the installer's offer and checking the release notes, just in case there's anything we missed or you should otherwise be aware of. Once you're happy, choose appropriate language and keyboard settings. You're then asked to install multimedia codecs, which you should do unless you have a reason not to (such as the proprietary licensing around some of them). Hopefully the Erase Disk And Install Linux Mint option fits your purposes. If not, choose Something Else and proceed with caution to the partitioning utility. We won't cover Logical Volume Management (LVM), but we encourage you to read up on it for



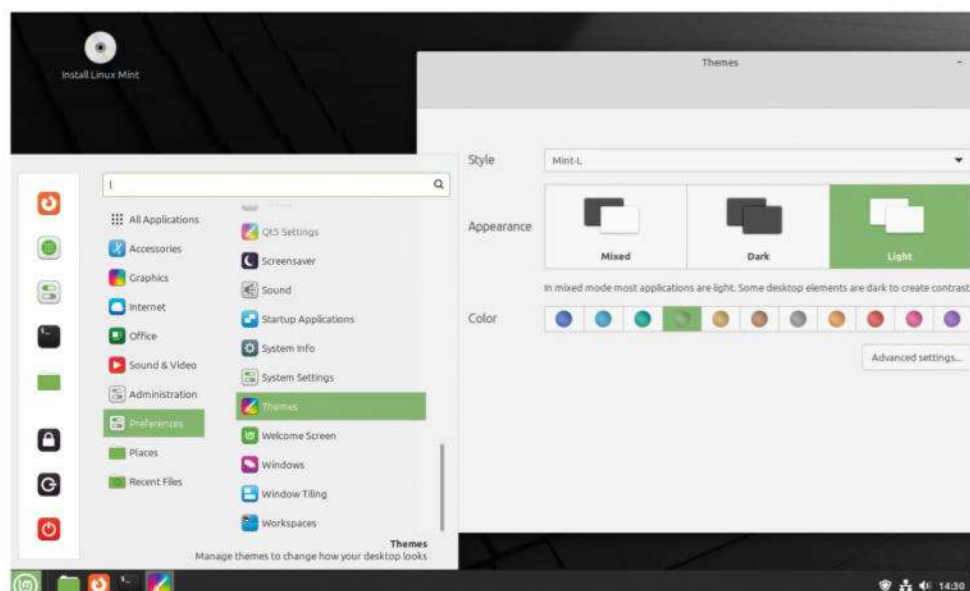
yourself (see Alex Burns's tips in **LXF317**). If you want to use it, tick the box in the Advanced features area.

When you're ready, hit Install Now and read over the summary carefully (in particular that the correct drive is targeted) before clicking Continue. Before the installer does its thing, you get to choose further localisation settings, as well as a username/password for yourself and a hostname for your machine. If you're installing on a laptop and are privacy-conscious, you might want to choose Encrypt My Home Folder. That way, any data stored there won't be compromised (without your password) in the event of theft. Data outside the **home** folder remains unencrypted. Encrypted data is much more susceptible to errors on the underlying filesystem, though, so always have a (secure off-site) backup of anything important.

Now at last begins the partitioning and copying of files. While this is going on, you can either enjoy the slideshow showcasing all of Mint's great features or make a cup of tea. When it's done, you're asked if you want to continue testing or reboot. Choose the latter, because we shall continue this journey over the page.

Mint comes with some absolutely dreamy backgrounds.

Adjust theme settings, including Light/Dark modes and highlight colours, from this screen.



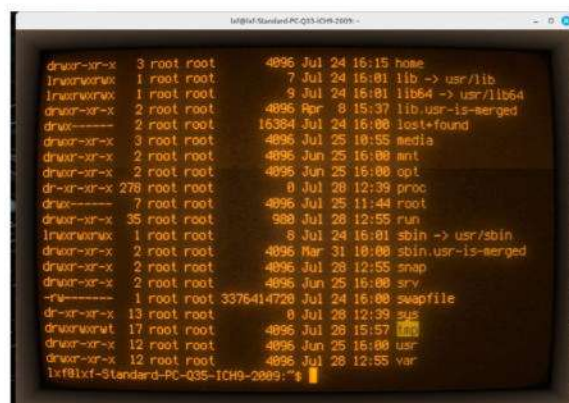
Getting to know Mint

Explore the freshest of fresh installs and tweak it to perfection.

Hopefully the machine reboots effortlessly to the Mint login screen (Lightdm) and after a flourish of credentials, you'll be back at the Cinnamon desktop. The first thing you'll see is a Welcome screen. As well as being a friendly greeting, this is actually quite useful because it guides you to all sorts of things you might want to set up, now that Mint is officially up and running. So useful that it pops up every time you log in until you tick the box (bottom-right) telling it not to. We'll cover a couple of things from the First Steps section, and leave you to follow through with the rest.

Any modern operating system needs to be updated, and we've already mentioned that Mint is no different. And that its update process is much less disruptive than you might find elsewhere. Be that as it may, sometimes an update goes wrong or contains a bug, and you wish you'd never applied it. In that case, you'd have to try to fix things manually, or wait for a new update to fix it. But with *Timeshift* (Mint's system snapshot tool), rolling back a rogue update becomes effortless. *Timeshift* is all about system packages, so it's not really for backing up your personal documents (although with some effort, you can make it do so).

Timeshift works similarly to Apple's *Time Machine* or Windows Restore Points. By sacrificing a chunk of disk space, you can use *Timeshift* to store incremental backups of the system folders on a daily, weekly or ad-hoc basis. This means the first backup is large (and slow), but subsequent runs use fancy checksumming and hashing to ensure only changes are stored. Our default install occupied about 13GB, and *Timeshift*



If you are as captivated by the terminal as we are, you really should check out Cool Retro Term.

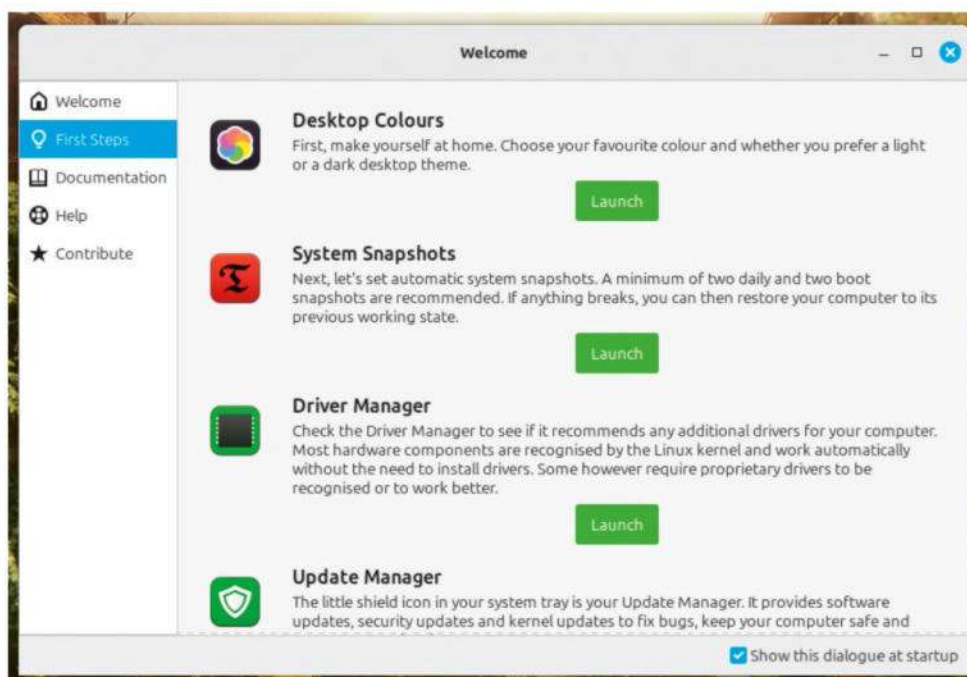
recommended we keep five daily backups. Unless we spent a week doing radical changes to the system, the total size of these snapshots would likely not exceed 20GB. A small sacrifice to avoid those nail-biting moments when the screen doesn't light up after a reboot. Even if your machine becomes unbootable, you can use boot the Mint Live CD and run *Timeshift* from there, pointing it at your install.

Run *Timeshift* from the Welcome wizard (or the Administration menu) and you're gently guided through the process. *Timeshift* does its best not to run you out of disk space (it doesn't make a new snapshot if there's less than a gigabyte free), but it's worth keeping an eye on. Running out of disk space causes all kinds of problems, including not being able to log in and fix

things. At least we're protected from update hell now. So, we may as well do our first system update. The *Update Manager* may already have popped up while you were fiddling with *Timeshift*. But if not (or if you dismissed it), you can find it in the First Steps section, or the Administration menu.

Fire it up and the first thing it tells you is that the updater wants to update itself. Spooky. Anyway, let it do its thing, and you're presented with a slightly arcane list of pending package updates. Before you apply them, heed the banner at the top imploring you to switch to a local mirror. We (in the UK) found that the main URLs (packages.linuxmint.com and archive.ubuntu.com) were actually fine, but things may be different elsewhere in the world. The mirror selection dialog will helpfully speedtest and rank other

The carefully thought-out welcome screen covers everything you might want to do in your first few Mint outings.



mirrors, so your updates can be even more speedy. Now, and after some password entry, you can apply the updates. If kernel packages (the real guts of Linux) are being updated, you might see a message about new packages being installed. Don't worry about this. These and other packages might also ask you to reboot in order to ingest them. You should probably do this, unless you're doing something really important.

In general, a home Linux user needn't worry about running antivirus or antimalware packages. That doesn't mean you shouldn't be vigilant about which sites you visit and which files you download. If you are running services (such as a web server) on your machine, or even if you're just paranoid that some malware is going to get you and run services on your machine, you can activate the *UFW (Uncomplicated Firewall)*. Being IPv4 babies, we've grown up with the (wrong) notion that being behind a home router acting as a NAT gateway (so everyone can share a single external IP address) protects us from unwanted incoming connections. Without going into why this is false (or at best incomplete), we are now (even in the UK) in the IPv6 age, and a great many things with microprocessors can all have their own address. *UFW* offers some easy presets to guard your packets, while at the same time not impeding day-to-day internet usage. So do give it a shot (especially if you're out and about on your laptop – public Wi-Fi networks are wild), either from the Welcome screen or Preferences menu.

In our testing, the *System Reports* tool decided to scare us by planting an exclamation in the system tray area. Investigating this, it claimed that reports needed our attention, so we duly clicked it. Only to be told that no problems were detected. Phew. You'll find the *System Reports* tool in the Administration menu (or when it springs a similar anticlimax on you as we just relayed). It's useful for gathering system information in case you need to diagnose a failure, ask for help in the forums or report a bug.

Free software shop

Now is a good time to introduce Mint's *Software Manager*, where you can seek out and download all manner of applications and games. You'll find it in the First Steps section of the Welcome wizard, or the second icon from the top in the launcher area of the main menu. Or in the Administration menu. Start it up, give it a moment to update its caches, and you're presented with some featured apps as well as a list of categories for hunting down others. *Software Manager* now has Flathub integration out of the box. So in the past where popular applications (such as *Blender*, *VLC*) would only have older versions available in the Ubuntu repositories, now you'll find brand new ones packaged (often by the developers themselves) and available as Flatpaks from the Flathub repo. Flatpaks, as well as being convenient and cross-platform, also offer confinement from other Flatpak applications and the rest of the system.

You can install whatever you like. If you want a free alternative to *Photoshop*, you should check out *GIMP (GNU Image Manipulation Program)*. If you want to play games, you should certainly check out *Supertuxkart*. You'll also find Steam and Lutris (the open source game library that supports Steam, GoG, Humble Store, Epic

» FIREFOX FEATURES

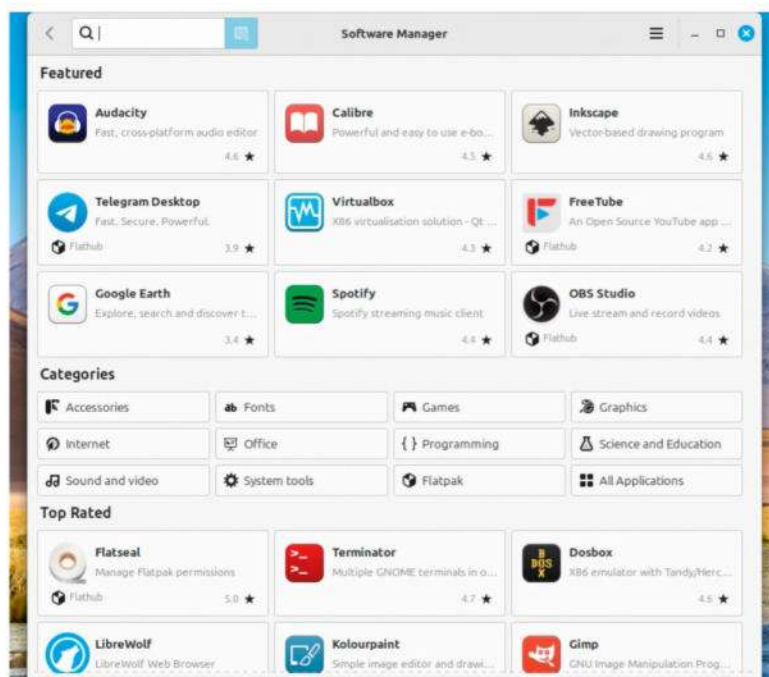
A great deal of people's desktop computing nowadays takes place in a web browser. So much so that it sometimes seems the underlying operating system is irrelevant. Be that as it may, Mint includes Mozilla's feature-packed *Firefox* web browser. You may be familiar with it from other operating systems, in which case you may already have kitted it out with your favourite extensions and themes. You might even have synced your bookmarks via your Mozilla account.

If you're new to *Firefox* (or even if you're not), we heartily recommend trying out its Reader Mode (by clicking the document in the address bar). It gets rid of all the extraneous bits of content on a web page, enabling you to focus on the actual article. If you really don't like ads (no one does, but people rely on them for income, so it's sometimes a moral quandary), *uBlock Origin* can banish them from your pages.

As a privacy-conscious publication, one final thing we'd implore you to do is change *Firefox*'s default search engine from Google to DuckDuckGo. The latter doesn't track you and is an open source favourite. Click the address bar, then click the cog in the lower-right of the drop-down menu to get to the search preferences. You'll find DuckDuckGo in the menu.

Games and more. Steam and Lutris both use a layer called *Wine* (sometimes Steam's custom version of it, called *Proton*), which enables Windows titles (particularly those using DirectX9 and later) to run seamlessly on Linux. This is what has enabled Valve's Linux-powered Steam Deck to be such a runaway success. See the box (above) for our rapid guide to pinging the *Firefox* browser. Or read on for some final tips and tweaks to get the most out of your install.

A treasure trove of free and open source software as well as a couple of proprietary tools are available in the Software Manager.



X marks the apps

Perfect your install and discover the new apps on the block.



If you want to transfer files from another machine on your local network, Mint includes the ever-so-useful *Warpinator*. You can download it for Windows (from <https://winpinator.swis.cz>) and it's available on other flavours of Linux. When you start it, it automatically detects other *Warpinator* instances on your network and allows you to instantly send files. You'll see a message warning you that its Secure Mode has been disabled. This means that *Warpinator* will stop listening for new connections in an hour, and that all transfers have to be manually accepted at the other end. To enable Secure Mode, you need to change the default Group Code (by clicking Set Code on the banner). Once done, update the code on other *Warpinator* instances and you have access to other options in the Preferences dialog.

If you'd like to defy Mint's anti-Snap stance (they don't like that the Snap Store is proprietary and controlled by Canonical), you can do so with a bit of command-line antics. This is unsupported and (unless you know some particular bit of software that you simply must have is only available from the Snap Store) you shouldn't need to do this. We just believe in freedom. Open the Administration menu and start the terminal. From here you can do all kinds of things to your system (including break it, if the `sudo` command is involved) with simple text commands. Delete the file that blocks the Snap Daemon being installed with:

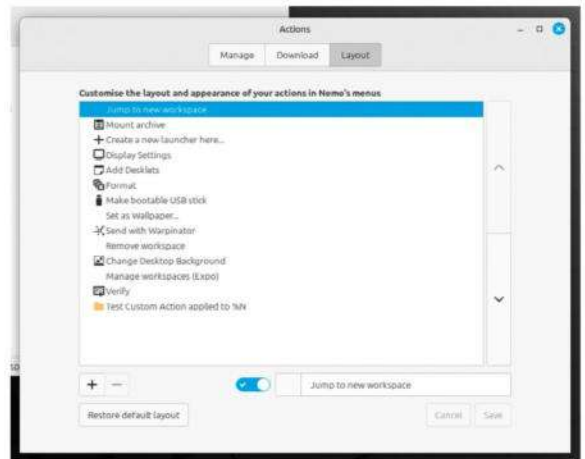
```
$ sudo rm /etc/apt/preferences.d/nosnap.pref
```

The `sudo` bit means the rest of the command is run as the superuser (so you're asked for a password). A regular user account can't delete system files. Update the package cache and install the Snap Daemon with:

```
$ sudo apt update
```

```
$ sudo apt install snapd
```

This won't give you the GUI Snap Store (only available in Ubuntu), but you can browse it at <https://>



There's now a layout editor for Actions in the Nemo file manager. You can download custom actions, too.

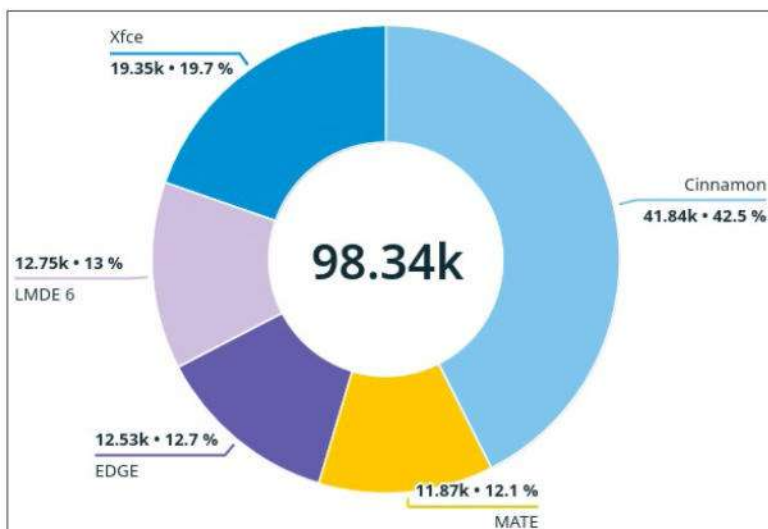
snapcraft.io and then use `snap install` from the terminal to install your desired selection. If you'd like to further defy Mint's careful design, you might want to enable unverified Flatpak apps in the *Software Manager*. Again, you shouldn't need to do this and Mint is right to have this off by default. The Flathub repository allows more or less anyone to upload any application with very little safeguards. Verified Flatpaks you can trust, others could contain malware, or just be terrible. Still, it's a popular request, and we'll happily enable you. Go to the hamburger (three stripes menu) in the *Software Manager* and select Preferences. The third option down enables unverified Flatpaks.

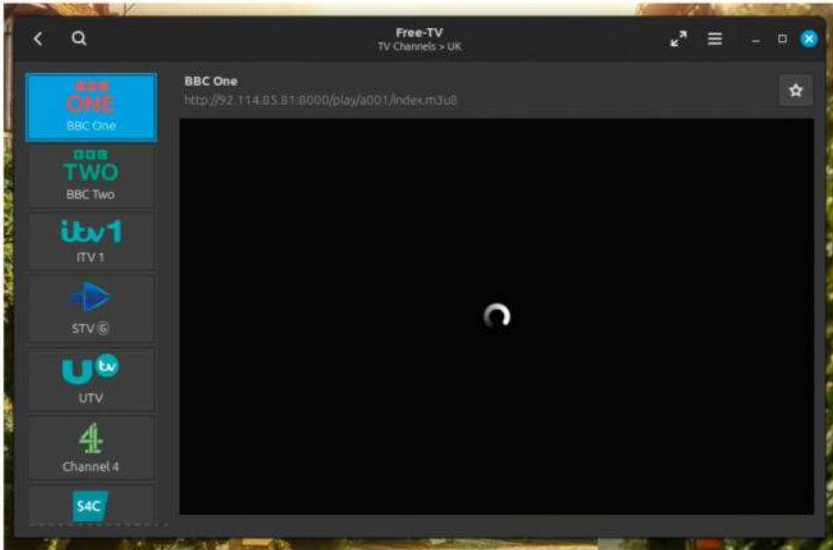
X-igent situation

The Mint team has been producing its XApp utility selection since 2016. The current line-up is *xed* (text editor), *xviewer* (image viewer), *xplayer* (media player), *xreader* (PDF and document reader) and *pix* (photo organisation tool). Oh and the Slick Greeter theme at login (and its control panel) as well as the Bluetooth front-end all fall under the XApp umbrella. The XApps came about because of certain applications becoming inextricably tied to the Gnome desktop. And either not working at all elsewhere or looking awfully out of place. So they forked the core apps and decoupled them from Gnome so they could work on other desktops.

Fast-forward to now, and Gnome's new theming model makes this effort seem all the more prescient. That theming model is based around a library called *libAdwaita*. *Adwaita* or *Advaita* in Sanskrit translates roughly as "non-secondness", "not two", or even more loosely, "the only one". It's a good name because the function of the library is to bestow the same *Adwaita* theme on all applications built using it. The upshot of this is that more and more formerly GTK applications are becoming, by virtue of adopting *libAdwaita*, Gnome

Thanks to a new partnership with DataDog, Mint can get detailed usage statistics.





Fans of IPTV streaming television will appreciate the bundled Hypnotix app and the access to thousands of free channels it brings.

apps. Other desktops (including Mint's Cinnamon), which use other theming models, can't touch these apps, which makes for a bad user experience. GTK is the low-level widget library that underlies about half of Linux GUI libraries and the Adwaita theme used to be built into it. With GTK4, though, it's been split off into its own library, which has caused something of a schism. Mint main man Clement Lefebvre made his feelings very clear about this in a blog post in April:

"LibAdwaita is for Gnome and Gnome only. We can't blame Gnome for this, they've been very clear about it from the start. It was made specifically for Gnome to have more freedom and build its own ecosystem without impacting GTK. We want to send a strong signal upstream and towards other projects. We cannot and will not support applications which do not support our users and environments. We can't promote applications to our users which don't support our users. The *Software Manager* will be vigilant towards that going forward and list compatible software by default."

And that's not just bluster. Linux Mint 22 has downgraded many of its applications (such as *Calculator* and *Calendar*) from GTK4 to GTK3, circumventing libAdwaita. Clem has also proposed making the XApp project independent of Linux Mint (outside of which it is rarely seen). By attracting outside effort, distros won't have to worry about not having suitably stylish core apps just because they don't use Gnome. We're already looking forward to more XApps, which Clem intends to fork from their GTK3 releases soon (as newer releases adopt libAdwaita).

There's actually another couple of XApp additions already. Notably the *Online Accounts* dialog is now part of the XApp cadre. You can use it to connect your Google and Microsoft accounts, or integrate your email with the desktop. Google Drive and Microsoft OneDrive storage then become visible in the *Nemo File Manager*. Of course, we think you shouldn't use these services and should be running your own Nextcloud server. Then you could connect that instead.

If you like kernel news [*who doesn't like Kernel News? – ed*], you'll be thrilled to hear that in another

break with Ubuntu, they will provide new kernels every six months. Vanilla Ubuntu will stick with the 6.8 kernel (continuing to maintain it with security fixes) for the whole five-year support period. Canonical does provide newer so-called Hardware Enablement (HWE) kernel images, just not by default. It's these that will be automatically rolled into Mint as they become available, ensuring cutting-edge support. That said, the 6.8 kernel will be fine for most people right now – it includes support for new AMD Zen 5 CPUs and Intel Xe graphics (both discrete and integrated devices).

One improvement we haven't yet covered is improvements for HiDPI users in the early boot process. This makes splash screen/console text look less ridiculous on, for example, a 4K display. In keeping with this new support, you might want to install a HiDPI theme for the boot menu with:

```
$ sudo apt install grub2-theme-mint-2k
```

Unsound advice

Sound problems on Linux are these days a rarity (printer problems, though, forgetaboutit). But if you do experience any audio aberrations, reverting PipeWire to the older PulseAudio sound server may help. This is easy and again involves the terminal:

```
$ apt purge pipewire pipewire-bin
$ systemctl enable --user pulseaudio
```

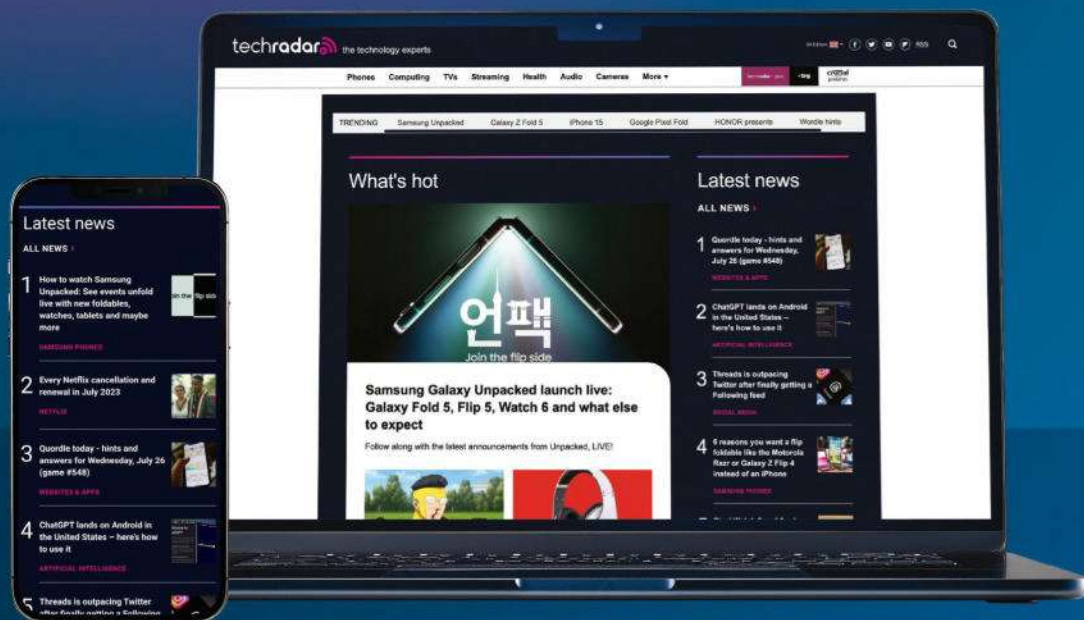
You need to reboot for the changes to take effect. And after that extravaganza, we need to reboot, too. We hope your Linux journey is a fragrant one. **LXF**

» GETTING HELP

Mint is easy to learn and extremely powerful, but sooner or later you'll get stuck. It's not Linux if you don't. Something will break or something will not make sense, and you'll find yourself looking for answers. It's always a good idea to Google... er DuckDuckGo, any error message that concerns you. Chances are someone's had the same problem, but don't blindly copy and paste random commands if you don't know what they do, especially from the first search result. It's always worth consulting the system logs in case there's any further information there (which you can also search on the web for). Run `journalctl -b` to see the journal for the current boot. Use the cursors (and PgUp/Down) to scroll through the records, and press `q` to quit.

You can do a lot worse than checking the official documentation (linked from the Welcome screen). Mint has some excellent package reviews on its community site (<https://community.linuxmint.com>). We've already mentioned the Matrix chat, but you can also share problems (or solutions) on the good ol' forums at <https://forums.linuxmint.com>. Do check your question hasn't already been asked. And do also familiarise yourself with the forum guidelines before you post anything. There is an art to asking a good question.

Meet the technology experts



- The world's **most** comprehensive technology website
- An **unrivalled** mix of news, opinions, reviews and features
- **All-new design**, new homepage, new features and special reports
- Backed by **over 300 years** of editorial experience

techradar
the technology experts

www.techradar.com

Pi Foundation releases AI teaching guidance

Worry no more about how to teach about ethical generative AI – the Pi Foundation has you covered.

For good or for worse, generative AI is here to stay. To help both students and educators navigate this tricky new area, the Pi Foundation, in conjunction with experts in the field, has produced guides on how best to understand the issues around AI, how to make use of AI in educational settings and how best to teach the subject.

A key aspect of the guide is educating students (specifically six lessons for 11 to 14-year olds) on the limitations of generative AI tools and exploring how they work through the use of *ChatGPT* and *Google Gemini*. The guide outlines the basic workings but also the pitfalls of using AI and will have students experiment with AI output and checking it for errors. Research continues on the best ways to

teach about AI, alongside guides on how to implement a school-wide policy in place for the use of generative AI.

Get links to the guide and a full explanation of what's covered here: <https://bit.ly/lxf319ai>



■ We all need guidelines now and again, even from AI itself...



Les Pounder works with groups such as the Raspberry Pi Foundation to help boost people's maker skills.

» GET YOURSELF CONNECTED!

Raspberry Pi recently announced a new product, but it's not a piece of hardware – it's software.

Raspberry Pi Connect (<https://www.raspberrypi.com/news/raspberry-pi-connect/>) is a VNC and SSH service for all models of Raspberry Pi. No longer do you need to install a VNC server on your device. Instead, *Raspberry Pi Connect* comes pre-installed on the latest Raspberry Pi OS and for older Pis you can use *Raspberry Pi Connect Lite* (even on the original Raspberry Pi from 2012).

So, how do we connect our PC to a Raspberry Pi running *Connect*? Open a browser, register your machines and select your Pi from the list. Your browser window becomes the Raspberry Pi OS desktop, and you can run apps on your Pi as though you are sitting there. Latency is low for devices on your home network; across the internet, it may get a little laggy. Unlike VNC, *Connect* can handle streaming video from the official Raspberry Pi Cameras. We tested both the terminal and Python camera libraries, and everything worked well. On a nice big screen, the user interface is clear and easy to use. On a mobile device, not so much, but *Connect* is still in beta and updates will fix this issue.

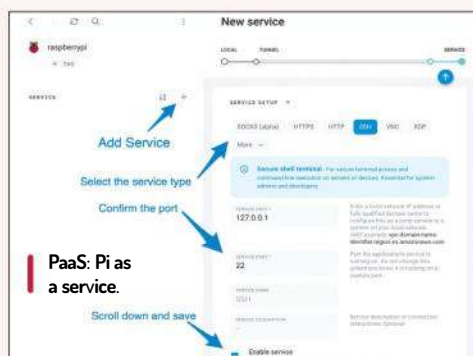
Raspberry Pi Connect will be useful in the classroom, enabling children to work on projects from various machines. Just think: developing code to control a robot from a laptop across the room. A live video feed on your screen and a custom user interface as you send the robot off to perform a series of tasks.

There is a full how-to on Tom's Hardware: <https://bit.ly/lxf319pi>

Remote it!

Bluetooth config tool.

Now you can remote headless install images to your Pi from any smartphone. Remote.it is offering an open source prebuilt image that, once booted on your Pi, offers remote install of Remote.it Pi packages and tailored services. It's an interesting service and one that could do a lot of heavy install lifting. Find out more: www.remote.it



CREDIT: Nagrom/Printables

Fractal build

Community takeover!

During Computex 2024, PC case maker Fractal Design showed off a miniature North case for the Pi, something many would buy in a heartbeat. The company doesn't have any plans to produce the case, it turns out, so in steps the Pi community! Several copycat designs you can 3D print yourself have appeared: <https://bit.ly/lxf319print>



■ We'll do it ourselves!

Fritzing 1.0.3

Les Pounder spends most of his work hours looking at circuit diagrams, then carries on doing the same thing after work.

IN BRIEF

For €8, *Fritzing* is a must-buy for makers and educators working with electronics such as the Arduino and Raspberry Pi. Providing a suite of applications to create breadboard and circuit designs, along with an extensive and expandable parts library, *Fritzing* is a jack-of-all-trades maker tool.

Chances are that you have seen circuit diagrams in *Linux Format* and on Tom's Hardware. They are made using *Fritzing*, an easy-to-use app for educators, learners and professionals to get their circuits designed. This isn't a full circuit design suite – for that you'll need *KiCAD* – but *Fritzing* is an important application in the maker community, and version 1.0.3 has just been released.

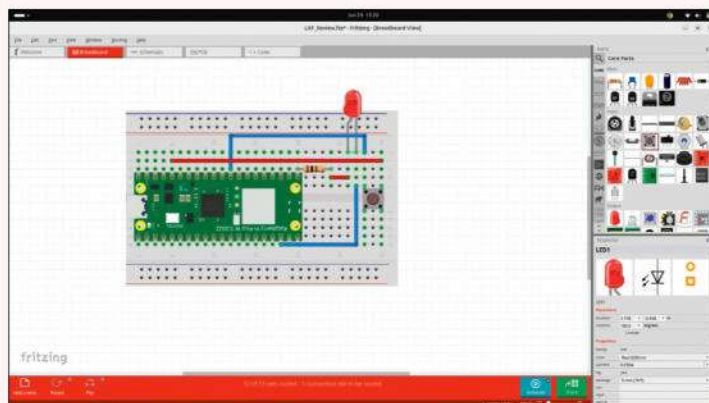
Let's get this out of the way: yes, you now have to pay for *Fritzing*. €8 is the smallest amount you can pay for continued access to future versions. This is a small price to pay for a quality product.

Fritzing is delivered as an ApplImage, so we have to make the file executable before we can use it. Once done, it runs really well on our Linux PC. There is no Raspberry Pi native version. Starting *Fritzing*, we are greeted by a Welcome screen, but there are tabs for breadboarding, circuit schematics, PCB design and even a code section for Arduino and Picaxe-based boards.

The breadboarding tab is where we spend most of our days. Here we can build a virtual electronics project using parts found in the Parts bin, at the top-right. There is a multitude of parts in there, but what if your favourite microcontroller isn't there? We can import individual parts, FZPX files and entire collections of parts created by third-party developers. We downloaded the official Raspberry Pi Pico W part, and installed it into the Parts manager. Then we used it to make a circuit, the one used in this month's Wokwi tutorial (page 44).

Back to the parts, and we have a library of parts on offer, all of which can be searched for, or you can scroll through thousands of parts. Each part can be inspected, changing its values. For example, we can specify the exact resistance value of a resistor, or the colour of an LED, down to the wavelength of the light that it emits.

Moving to the Schematic tab, we had a schematic of the same circuit, but it did have a few errors. The errors



The breadboard section is where we spend 95% of our time.

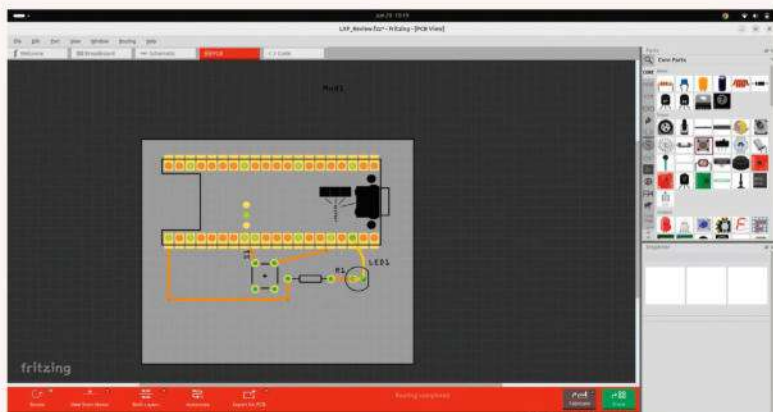
required us to move parts around to make them more understandable. No big issues, but could be a gotcha for your electronics classes.

What about designing your first printed circuit board (PCB)? *Fritzing* can do that. It may not be the cool kid in class, but *Fritzing* can design circuit boards. Drag your components around the PCB and tweak the connections and routes to get it right, add a silkscreen print to the design, and send it off to a service for fabrication.

You've got the design, but what about the list of parts? The Bill of Materials (BOM) can be exported as an HTML file or as a CSV for those who love spreadsheets, and for the savvy maker who wants to share their projects with ease.

The beauty of *Fritzing* is that we can use it just like *Inkscape*, but for electronics. In fact, we can export designs as SVG or other popular image formats. That means we can tweak the diagram for use in print and online. For breadboard diagrams, like those used in *Linux Format*, *Fritzing* is superb. It does have an annoying habit of not allowing us to connect the dots between breadboard components, but this is very rare. On the whole, it just works. For PCB design, it can be used to get you started, but for serious work, learn *KiCAD*. **LXF**

You can make PCBs ready for fabrication with *Fritzing*, but a more advanced tool is needed as you grow your skills.



VERDICT

DEVELOPER: Fritzing Team

WEB: <https://fritzing.org>

PRICE: €8

FEATURES 8/10

PERFORMANCE 8/10

EASE OF USE 9/10

VALUE 8/10

For €8 you get a great application to help document and share your circuits. It's not perfect, but does everything well.

» **Rating 8/10**

Sovol SV08

Big, super-fast and open source, it looks as though **Denise Bertacchi** has found her perfect printer!

SPECS

Build: 350x350x345mm

Type: PLA, PETG, TPU, ABS (up to 300°C)

Extruder: Direct drive, planetary gear
Nozzle: 0.4mm steel Bambu
Bed: PEI spring steel sheet, heated, auto-levelling

Sensor: Runout, inductive, pressure, camera

Comms: USB, Ethernet, Wi-Fi

Control: 5-inch LED screen with knob/Mainsail

Size: 400x400x530mm, 13.3kg

The Voron Design team is a volunteer group of engineers who create free open source blueprints for the best-quality 3D printer you can print, build and modify yourself. Some manufacturers have been accused of helping themselves to Voron's designs to make shameless clones. But not Sovol. This is plainly labelled as a mass-manufactured version of the Voron 2.4, giving you the quality and modability of Voron's design with the affordability and simplicity of a ready-made product. Sovol reached out to Voron Design and asked to do a partnership with a sponsor deal; for every SV08 purchased, it puts \$2 in the Voron Design tip jar. As of this review, they've donated over \$2,000.

The Sovol SV08 has an incredible top speed of 700mm/s and a blistering max acceleration rate of 40,000mm/s². This gives it the ability to top our list of fastest 3D printers with a 13-minute and 25-second speed Benchy, making it one of the best 3D printers on the market right now. It's also a beast of a machine, with a 350x350x345 build volume, and four independent stepper motors at each corner of the flying gantry.

The Sovol SV08 has a hard-mounted bed that doesn't move and a flying gantry for the X and Y axis taken from the Voron 2.4 design. Four motors on each corner of the gantry, combined with an inductive probe, enable it to quad level for a perfect first layer. The axes are belt-driven for buttery smooth layers. The hotend is beefy, with an all-in-one nozzle, massive cooling fan and convenient daughterboard for the electronics.

The printer has inductive sensors for hands-free bed-levelling, a pressure sensor on the nozzle for finding the Z offset, and an accelerometer for input shaping. It can perform a quad Z tilt, where it automatically adjusts all four Z axes to tram the bed on its own.

Using vanilla *Klipper* allows easy Wi-Fi access to the printer and its files – there's no need for the cloud. You can also send files via a USB flash drive or plug right into your network with an Ethernet cable. The firmware is already set up with Crowsnest to automatically find and display the SV08's gantry mounted camera.

The Sovol SV08 is a kit that takes about an hour to assemble. Though this is more work than many of



today's printers, it is extremely easy compared to a built-from-scratch Voron.

The supplied *Orca Slicer* has a preset for the SV08 baked in, but you can also use *Cura* or *PrusaSlicer* and the default Voron settings. We found the settings on all three slicers to be extremely conservative, so you may want to do a little experimenting to reach faster prints.

The Sovol SV08 printed great right out of the box. Our first print was a Benchy using the sample spool that was more generous than most. The boat is a bit rough, but the shape is well defined with no signs of ringing. It's one of the best speed boats we've printed. Remember, when racing a printer, it's more about printing a Benchy-shaped object than real quality. It was printed in ordinary grey Inland PLA, so none of the defects are hidden.

For less than £600, the Sovol SV08 is a great choice for someone who wants an extra-large build volume and Voron speed but doesn't want to build a printer from scratch. This is very much a finished piece of hardware, but you can still tinker and customise it as you like. **LXF**



3D Benchy was done in record time and looks pretty decent, too.

VERDICT

DEVELOPER: Sovol

WEB: www.sovol3d.com

PRICE: £525

FEATURES	9/10	EASE OF USE	7/10
PERFORMANCE	10/10	VALUE	8/10

A big, fast, single-colour machine without the need to send files over the cloud. It's open source, which means you're also able to upgrade and mod the printer as you see fit.

» **Rating 8/10**

WOKWI

Credit: <https://wokwi.com>

Simulate your Pi Pico projects with Wokwi

Claiming he's saving resources, **Les Pounder** simulates an electronics project because he is too lazy to get up and build it for real.



OUR EXPERT

Les Pounder is associate editor at Tom's Hardware and a freelance maker for hire. He blogs about his projects at <http://bigLes>.

YOU NEED

- > **Pi Pico or Pico W**
- > **A half-size breadboard**
- > **A 100 ohm resistor (brown-black-brown-gold)**
- > **3x M2M jumper wires**
- > **A push-button**
- > **Code:** <https://github.com/lesp/LXF319-Simulating-Circuits-with-Wokwi/archive/refs/heads/main.zip>

Sometimes we want to test electronics projects without having access to the technology. We're on a train, waiting at the airport, or in the classroom, sharing a kit. For these occasions, we need a simulator and we have a great one with Wokwi.

Wokwi is a free, online circuit simulator where we can create projects for the Raspberry Pi Pico, STM32, Arduino and ESP32. The simulator provides an environment to build a simulated circuit and write the code to control it.

In this tutorial, we will create an example circuit for the Raspberry Pi Pico and write some MicroPython code to control it.

We'll then port the code to a real Raspberry Pi Pico and show the same circuit running on real hardware.

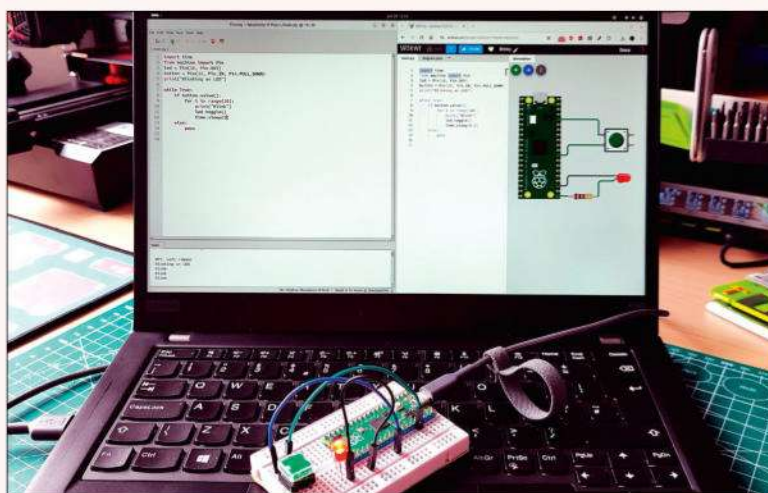
Open a browser and go to <https://wokwi.com>. Click on the Pi Pico icon, scroll down to the Starter Template and select MicroPython. Do not select Pi Pico or Pi Pico W, because these run Arduino code examples.

The Wokwi simulation interface is split in two: a coding section and the simulation. In the simulation, we can build the electronic circuit. This is where we'll start.

We start with just a Raspberry Pi Pico. To add components, click on the + icon at the top of the simulation section. From the list, select LED to drop an LED into the simulator. Move the LED down to the bottom-right of the Pico. Look to the top of the simulator and you will see different colour options, plus mirror, rotate and delete icons. With the LED selected, click on the rotate icon and the LED will rotate 90 degrees clockwise.

The next component is a resistor. Click on the + icon and select Resistor. Drop a resistor into the simulator and with the resistor selected, change its value to 100 ohms (brown-black-brown-gold). Rotate the resistor so that it is horizontal and then drag it down to the bottom-right of the Pico.

We have two components floating in the simulator – let's connect them to the Pico. The bottom-right GPIO



Two identical circuits: one is a simulation, the other is very real. Both work exactly the same way.

pin of the Pico is GPIO16. Click on the pin and a green wire appears. You don't need to hold the mouse button down, but drag the wire to the left leg of the resistor. Click on the green square that appears on the resistor. You've just made a connection. Now do the same from the right leg of the resistor to the crooked leg of the LED (anode). The final connection for the LED sees the remaining leg (cathode) connected to a GND pin, two pins up from GPIO16.

We've got an output, so let's now add an input. A button (momentary switch) is the ideal introduction to inputs. From the + icon, select PushButton and you will see a huge button appear in the simulator. Drag the button so that it is at the top-right of the Pico. We need to connect the top-left leg of the button to the 3V3 pin of the Pico. Hover the mouse over the top-right GPIO pin of the Pico. Now move the mouse down the pins and you will see a pin number/function appear. Look for 3V3 (not 3V3_EN), which is five pins down. Make a connection from this pin to the top-left leg of the button. Now, from the bottom-left leg of the button, make a connection to GPIO22 on the Raspberry Pi Pico. Remember to hover the mouse over the pins to find GPIO22. You may notice that the wire is hidden or obscured. We can fix this. Click on the wire and you will see an animation play. More importantly, there are a

few purple dots. Click on the dots and move the wire away from the Pico.

We've built the circuit, but right now it doesn't do anything. That's because we need to write some MicroPython code.

On the left side of the interface, we have a MicroPython editor. Highlight and delete any code in there. We'll start afresh.

Import two modules: time and, from the machine module, Pin. We'll use time to add delays to our code, while Pin is used to control and read GPIO pins:

```
import time
from machine import Pin
```

Now we create two objects that refer to the LED at GPIO16 and the button at GPIO22. The LED is a simple output device, so we tell MicroPython to set GPIO16 to an output – this will send current to the LED via the resistor. Because the other leg of the LED is connected to GND, an electrical circuit is completed.

```
led = Pin(16, Pin.OUT)
```

The **button** object is an input. We press the button and connect the 3V3 pin to GPIO22, changing the state from **LOW** to **HIGH**. The low state is achieved by telling MicroPython to pull the pin down:

```
button = Pin(22, Pin.IN, Pin.PULL_DOWN)
```

Add a quick **print** function to tell the user that the code is running:

```
print("Blinking an LED")
```

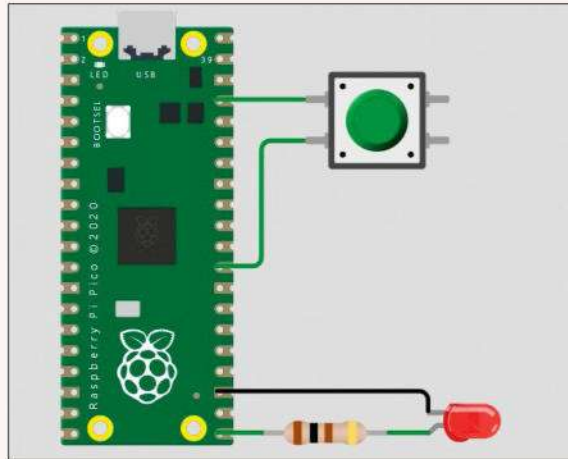
The main loop is a **while True**. It runs a conditional test that checks for a button press. If the **button.value()** changes from **LOW** to **HIGH** (**False** to **True**, **0** to **1**), the code triggers:

```
while True:
    if button.value():
```

What runs is a **for** loop that iterates ten times. Each time it loops, it prints "Blink" to the Python shell, then uses a toggle to turn the LED on and off. A sleep of 0.2 seconds is enough to see the LED change state.

```
    for i in range(10):
        print("Blink")
        led.toggle()
        time.sleep(0.2)
```

If the user doesn't press a button, we want the loop to keep running. Using an **else** condition, we just pass the code so that the main loop repeats:



Wokwi's simulated circuit for this project can be used as a reference to build a real circuit on a breadboard.

```
else:
```

```
    pass
```

Click on Save and name your project. We chose **LXF_Blinky**, but feel free to change the name of your project. The project can be public or unlisted. Choose as you wish. Click Save when done.

In the simulator, click on the play button to start the code. In the bottom-right is a Python shell. It should show "Blinking an LED". Now click on the push-button in the simulator. The LED blinks on and off!

You've just created a simulated circuit and controlled it with real MicroPython code. Now let's try it on real hardware.

In real life, set up your Raspberry Pi Pico circuit to match the simulation (there's a high-resolution circuit diagram in the download). You need *Thonny* installed on your machine – we cover that in the boxout (below).

In Wokwi, click on the down arrow next to Save. Download the Project ZIP and extract the contents. In *Thonny*, click on File > Open and select This Computer. Navigate to where the extracted files are located, and select the **main.py** file. Click File > Save As and save the file as **main.py** to the Raspberry Pi Pico. Click on the green Run arrow to start the code. The Python shell will show the "Blinking an LED" message. Press the real push-button to blink the real LED.

You've just created a real project from the simulated project code and circuit design. **LXF**

» INSTALLING THONNY

While holding down the BOOTSEL button, connect your Raspberry Pi Pico to your computer. Go to www.raspberrypi.com/documentation/microcontrollers/micropython.html#drag-and-drop-micropython and download the version of MicroPython for your Pico or Pico W. Open your file

manager, then go to the downloaded file and copy it to the root of the **RPI-RP2** drive. This flashes the new firmware to the Pico.

Using your distro's package manager, install *Thonny*. For the latest Ubuntu release, we used a Snap package:

```
$ sudo snap install thonny
```

Open *Thonny* and connect the Pico to your machine. Go

to Tools > Options and select the Interpreter tab. Set the interpreter to MicroPython (Raspberry Pi Pico) and set the Port to match the location of your Pico. Click OK. *Thonny* now connects to the board and we can start writing code.

Thonny is our preferred Python editor these days. Capable of writing Python, MicroPython and

CircuitPython, this free app is designed for learners and eager makers. It has a package manager for Python and MicroPython, so we can easily install modules without using *pip*. *Thonny* has been the default Python editor for a few years now, moving away from the *IDLE* Python editor. *Thonny* is easier to use and very user-friendly.

» GET YOUR Pi FILLING HERE Subscribe now at <http://bit.ly/LinuxFormat>

Part four!
Don't miss
next issue,
subscribe on
page 16!

PIPEWIRE

Matt Holder delves into the depths of our audio stack to learn about PipeWire and what came before it.

In parts one and two of this *Inside Linux* series, we covered the Linux kernel and learned all about the different layers and subsystems that enable us to interact with all of the hardware connected to our computers. In part three we looked at graphical displays and how Wayland has been developed to replace the venerable X server.

The aim of this fourth part is to help you understand how the audio subsystems on your Linux distro work together, to give you a professional grade system that can be used for pretty much any purpose.

Once we have introduced the various systems, we will move on to describe how PipeWire can be used to provide some of the missing functionality from Wayland (such as the capture of apps and displays).

The Open Sound System (OSS) project started over 30 years ago and was developed for use on both BSD and Linux. The code was released under multiple licences, and the final release of OSS was in 2019, but it was superseded by the turn of the century on most Linux distros.

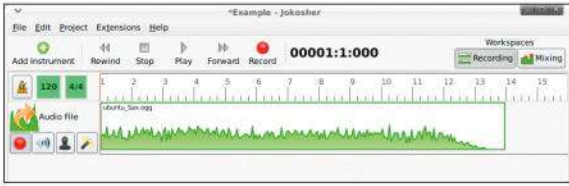
Get us ALSA here

Due to perceived issues with OSS, the Advanced Linux Sound Architecture (ALSA) project was first released in 1998. Until 2002, it was developed as a standalone project and then became part of the Linux kernel. In a similar way to OSS, ALSA provides a device driver between the kernel and userspace applications. The ALSA API is more complicated than OSS, which can make the creation of apps more complicated (although these days, this issue is abstracted to the graphical toolkit being used and the sound servers we will soon discuss). At conception, ALSA supported more features than OSS (such as hardware mixing of multiple channels and thread-safe drivers). Continued development of OSS likely means that the feature sets supported by both systems are largely the same.

Both OSS and ALSA only support a single application being able to interact with the audio hardware at any one time. This is very limiting – imagine not being able to take part in a Teams call in



CREDIT: Magictorch



The Jokosher multitrack editor was developed using GStreamer and we vaguely remember discussions around support being added for geographically separated instruments – which sadly never happened.

which we wish to share our screen and play a video for the other members of the meeting to see, or being able to play music on our favourite streaming service while still being able to hear notifications from our applications. To counteract this limitation, sound servers have been developed, which sit on top of ALSA. Applications therefore interact with the sound server, which in turn mixes any audio sources together, routes them to different applications and then, where needed, interacts with the hardware via ALSA.

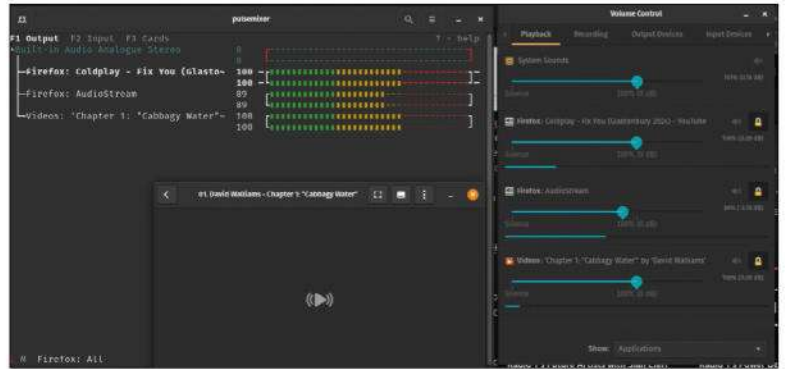
Poettering around...

Lennart Poettering is a prolific developer in the open source community. For many years, he worked for Red Hat, before moving to Microsoft in 2022. He worked on the open source version of Apple's Bonjour, called Avahi, which allows devices and software to advertise themselves on the local network to be discovered by other devices. Poettering's current project is the amazingly complex *systemd*, an init system that controls the running of all services and processes on our Linux distros.

Before Poettering worked on *systemd*, he spent a lot of his time working on PulseAudio. This is a sound server, which allows applications to interact with it and then handles any mixing or routing that is required, before interacting with ALSA, which in turn interacts with the kernel. Unfortunately, when PulseAudio was first adopted, there were lots of issues, which left users very frustrated. Whatever the cause, it seemed as though we took a step backwards. However, time flies and those issues with first integrating PulseAudio occurred around 16 years ago. As well as being able to support multiple devices playing audio concurrently, PulseAudio has support for virtual devices, which can be used to route particular sources to different apps, and also has support for network audio streams.

Jack it all in!

The next sound server to discuss is JACK Audio Connection Kit. JACK caters for the professional user who needs to control every element of their audio stack. JACK offers an amazing array of features, including the ability to use a patch bay approach to routing audio between apps. For example, you may want to take the output of one app and send it to be the input of two other programs, while sending audio from another process to an audio recording application. All of this is possible, as well as much more. JACK could be the perfect sound server for people who wish to produce professional audio recordings



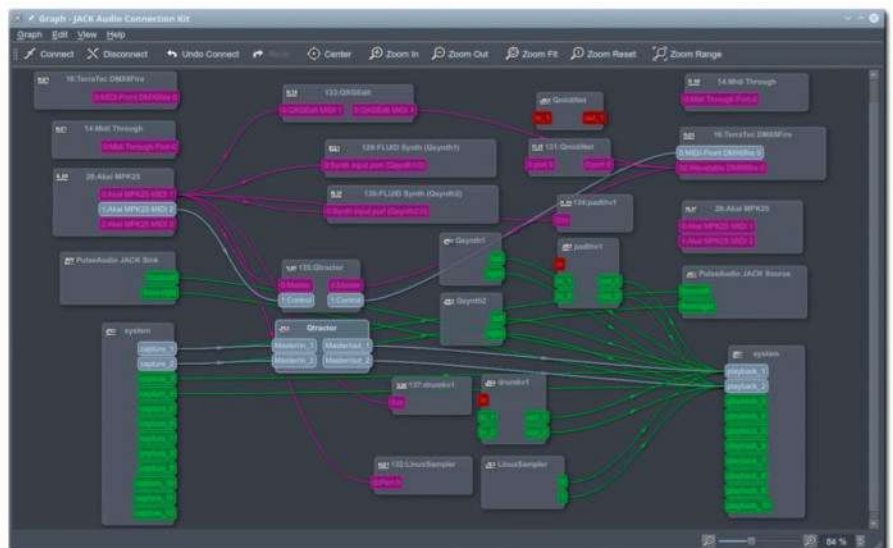
PulseMixer and Pavucontrol are both running here and show three different audio sources from which PulseAudio is accepting audio.

for bands, or the keen musician who wants to route their multi-channel audio interface into a digital audio workstation (DAW), as well as using an open source app to create drum loops. All of these sources can be fed into a DAW, such as *Ardour*, which can be used to record each track separately. Another use of JACK could be to allow podcasters to record audio from local and remote users.

The PipeWire project began in 2015 as the video equivalent of what PulseAudio did for audio in Linux. In 2017, the main developer, Wim Taymans, began to add audio functionality. With the aim of supporting both consumer and professional audio users, he consulted with a JACK developer and an *Ardour* developer. It could be argued that PipeWire is revolutionising the audio/visual experience on our Linux systems, as it aims to provide capture and playback of audio and video with minimal latency, real-time processing of multimedia sources, support for PulseAudio and JACK, and a robust security model, which allows seamless support for containerised applications, such as Flatpak.

Compatibility with the PulseAudio and JACK source servers has been handled in a really clever way, where components have been written that are compatible with both systems. This means that PulseAudio has been swapped with pipewire-pulse and the distro isn't aware that anything has changed. Given the teething problems when PulseAudio was first adopted, this approach means it won't happen again. JACK support is handled in a very similar way. The screenshot (above) shows a PulseAudio volume control running on Pop!_OS. Interestingly, Pop!_OS is running PipeWire and

QjackCtl can be used as a patch bay to route audio between different applications.





OBS with a simple configuration allows for audio and video capture from multiple sources.

pipewire-pulse, and our OS cannot tell the difference. As we embark on a containerised future, having proper support for piping audio/video into and out of our apps is really important and this support is all built in.

The final piece of the puzzle is the session manager, which provides an intermediary layer between the applications and PipeWire itself. See the interview with George Kiagiadakis (*opposite page*), main developer of WirePlumber, PipeWire's session manager.

Last month we discussed Wayland and how it is replacing the venerable X server. One problem that was present in the early days of Wayland was that there was no way to provide remote access from outside of the network, take screenshots or record the screen to create videos for training or other purposes. The Wayland specification doesn't disallow these features, but nothing was built into the reference implementation, which is called Weston.

While this paragraph is not strictly relevant to PipeWire, it is useful to know that when it comes to providing remote support, *TeamViewer*, *RustDesk* and *SimpleHelp* all offer a level of support for the Wayland display server. This generally means that remote support sessions are allowed, where a user allows the session request, but unattended support does not seem to be possible. For accessing your own system, KDE Plasma has gained support for Microsoft's Remote Desktop Protocol (RDP). By ticking a box in the Desktop settings, you can enable the functionality.

Lights, camera, action!

We will now take a look at the excellent *OBS Studio* (<https://obsproject.com>) which can be used to record our desktops. It's available as a standard Flatpak and has install support for Ubuntu, Debian, Fedora, and Arch. See the download page for details.

OBS has support for various audio and video sources, as well as different options to record our desktop. Let's first of all take a look at the main interface and then we will install the software, before configuring it to record various things – you can find a full guide to *OBS Studio* in **LXF293**.

The screenshot (*left*) shows the main screen of OBS, and the central rectangle provides a preview of the sources that have been selected. The bottom-left panel shows a list of scenes and each scene can have different sources added to it. Next to the Scenes panel is the sources that have been added to the selected scene. The next panel along shows the Audio Mixer. The penultimate panel on the bottom section is Scene Transitions, which provides a nice way of adding transitions between scenes. Finally, we see a panel containing buttons to open the settings, open the studio mode and to start streaming and/or recording.

We'll mock a steamer's style, where we will record the screen, webcam and the built-in microphone.

If you've not installed *OBS Studio* yet, install it from your package manager and open it. Next, complete the first-run wizard to configure some sensible defaults.

Make sure that you have a game installed that you want to record, then from the *OBS Studio* screen, use the + button in the Sources panel to add a Window Capture (PipeWire) source. This asks you to select which application you want to capture, so make sure the game is running before you add the source. If the game is full-screen, you may need to use Screen Capture (PipeWire). Next, add another source for Video Capture Device. The PipeWire source for video devices is currently in beta, so this may or may not work for you and you may need to use the V4L version (our laptop webcam wasn't supported by the PipeWire video capture source). Whichever works for you, select the source you want to capture and accept the options on the dialog box to add it to your scene. Use the preview window and the drag handles to change the size and position of each source.

Take a look at the Audio Mixer panel. By default, you have two options: one captures the desktop's audio and the second your webcam's mic. Experiment with the levels from these two sources so that the game's audio is recorded as well as any commentary that you wish to record. If you wish to add another audio source, such as an external microphone or audio capture device, you can do this in the Sources panel. Notice how the Audio Input Capture and Audio Output Capture sources are suffixed with (PulseAudio). Due to the previously discussed compatibility layer, these sources do still use PipeWire behind the scenes.

Referring back to the screenshot, you can see how your screen should look when using various PipeWire and PulseAudio capture sources from within *OBS Studio*. Experiment with the Start/Stop Recording functionality to record yourself playing the game and check that the audio levels are correct. By using the Stream tab in Settings, you can configure plenty of streaming sources to send our live game-playing and commentary to Twitch, YouTube or other sources.

We hope that you've found this series of articles interesting and can see what a brilliant position we are now in with incredible technologies, such as the Linux kernel, Wayland for our graphical display, and PipeWire for multimedia processing. We always find it interesting to think back to the state of things when we first started to use Linux. We have come such a long way since then, and to have a free operating system that automatically recognises and configures most hardware is truly amazing. **LXF**



» INTERVIEW WITH GEORGE KIAGIADAKIS

George Kiagiadakis is a principal software engineer at Collabora, working on the multimedia domain, and is the maintainer of WirePlumber. We sat down for a chat with George about all things PipeWire.

Linux Format: What interests you about the audio/visual subsystems used by Linux distros?

George Kiagiadakis: As an engineer, I have always been captivated by the complexities of implementing software, and multimedia software presents an especially intriguing challenge. But aside from that, the Linux audio/video frameworks and subsystems also have the potential to provide solutions for a number of use cases that go far beyond the typical Linux desktop environment. In my professional work, as a consultant at Collabora, this allows me to help our customers build very interesting products based on this software, while contributing much-wanted functionality to the Linux desktop community, thanks to all of this work being open source. This combination of challenge, opportunity and balance has been my motivation for years.

LXF: From a technical perspective, can you discuss the differences between PulseAudio, JACK and PipeWire?

GK: PulseAudio (PA) and JACK have a fundamental user experience difference, as PA offers a solution that is easy to use, with minimal configuration and a lot of automation, while JACK is tailored for professionals who do audio production with complex setups, requiring detailed manual control and flexibility. But there is also a technical difference underneath. PulseAudio is implemented in a way that doesn't work well with small buffers, so it can't achieve very low latency. It requires buffering tens of milliseconds worth of audio, delivering it all at once to the output. This is different from JACK, where an efficient

scheduling mechanism allows it to work with small buffers and achieve very low latencies, which is extremely important in professional audio setups. Now, PipeWire under the hood is very much like JACK. It has similar mechanisms that allow it to achieve very low latencies. However, it differs from JACK in that it is not limited to professional audio as a target use case, so many additional support mechanisms have been built in, allowing PulseAudio's user experience to be implemented on top. For instance, an important support mechanism for this is the audio converter that transparently converts between audio formats and sample rates for non-JACK applications.

LXF: What are the most important things that have been added to PipeWire to allow it to be a part of our everyday workhorses?

GK: First and foremost, it's PipeWire with PulseAudio that has made it

offer an easy and efficient way to bridge audio to and from JACK, so in cases where these features are crucial, it is possible to run the two systems alongside each other, seamlessly.

LXF: How widespread is the usage of PipeWire across the main distros now?

GK: Oh, it is the default audio system everywhere! I am not aware of any recent distribution that still uses PulseAudio as the default. It is also the default means of implementing screen sharing on Wayland, which is also very widespread among distributions and desktop environments.

LXF: How do the capabilities of PipeWire compare with that of the audio systems in other proprietary operating systems?

GK: PipeWire is much more advanced than any proprietary audio system. Windows and Mac OS audio systems, of course, aim at a good user experience and in many aspects

JACKED UP

“PipeWire under the hood is very much like JACK. However, it is not limited to professional audio, so many additional support mechanisms have been built in.”

successful. I remember I was discussing with a colleague once and he was telling me he was still using PulseAudio as he hadn't found the time yet to upgrade his setup, but when he eventually looked, he found out that he had been using PipeWire for months without noticing. Another thing worth noting also is the awesome Bluetooth audio integration, which makes using Bluetooth audio equipment a very pleasant experience.

LXF: Are there any other areas where PipeWire is lacking, compared with other Linux audio stacks?

GK: Yes, there are still some niche features in JACK that are not implemented in PipeWire. I am not very familiar with them myself, as I have never worked with such complex pro audio setups. However, PipeWire does

resemble PulseAudio and by extension PipeWire, but they are much more limited in functionality and extensibility. In some cases, their business logic even fails at basics, at least in my experience with Mac OS...

LXF: Can you tell us a bit about yourself please

GK: You might know me as the author and maintainer of WirePlumber, PipeWire's session manager, but I am a principal software engineer at Collabora, on the multimedia domain, and aside from PipeWire, I have done a lot of work also with GStreamer and other projects. I have been involved in open source for more than 15 years and I deeply care about a lot of projects and the community around them. I was born and raised in Greece and still live there, working remotely.

Try a tiny text editor

Shashank Sharma lives by the try-it-before-you-dismiss-it code. That's how he finds new tools. It's also how he discovered a fondness for kimchi.



**OUR
EXPERT**

**Shashank
Sharma**
is a trial lawyer
in Delhi and an
avid Arch user.

Manchester United, M Night Shyamalan, cargo trousers and your favourite text editor all have one thing in common: they all failed to achieve universal appeal, which as a concept is unattainable. You'll sooner be able to settle the debate about whether Han Solo fired first or Greedo than unite the universe into liking the same thing. While the '90s and '00s were spent raging about *Emacs* and *Vim*, our troubled Linux distros have even more text editors vying for attention now.

One such text editor is *Micro*. If you've used *Nano*, you can guess from the name that *Micro* aims to follow in the same footsteps. Designed to be lightweight, *Micro* boasts of extensive mouse-support and useful default

```
1: linuxlala@playground: ~/config/micro/backups
1: RFA 370/2016
2: ref no. 2510767/2024
3: court fees - DLCT2543624330761
4:
5: RFA No. 389/2016- For Respondent
6: ref no. 2510867/2024
7: court fees - DLCT2534624360011
8:
9: RFA No. 393/2016- For Respondent
10: ref no. 2511167/2024
11: court fees - DLCT2501624420834
12:
13: RFA No. 395/2016- For Respondent
14: ref no. 2511287/2024
15: court fees - DLCT2524624460537
16:
17: RFA No. 415/2014- For Respondent
18: ref no. 2511427/2024
19: court fees - DLCT2541624490835
20:
21: RFA No. 555/2014- for Appellant
22: ref no. 2511587/2024
23: court fees - DLCT2542624530934
/home/linuxlala/Desktop/UFLX RFA inspection-det No name + (12,1) | ft:unknown | unix | utf-8Alt
+ replaceall Respondent Opp. Party

1 Additional Information:
2 Application Form signed - payment Rs. 30 lakh
3 BBA signed - total payment Rs. 60 Lakhs.
4 Villa consist of Basement, Ground Floor
5 + 2 Floors on the Plot.
6
7 Customer opted CLP (Construction Linked Plan)
8 to pay the installments.
9 Possession within 30 months from the
10 commencement of development work.
11
12 i.e. September 2014
13 Total payment made Rs. 7.43 Crore
14 out of consideration amount of Rs. 8 Cr.
15 All payments made between 29.11.2011
16 to 19.04.2017
17 In March 2014, the customer visited
18 the Villa and found that Vastu orientation
19 of Villa no. 69 had been reversed.
20 Customer communicated through mails dated 03.
21 In his mail dated 15.04.2014, the complainant
22 IOP sent on 17.01.2018.
23
```

A simple, easily extendible and configurable text editor that you can't ignore out of hand.

keybindings. It also features persistent undo, the ability to split the interface vertically and horizontally, replace words, highlight search results and more. Even better, you can get additional features through plugins, which can easily be deployed with the built-in plugin manager.

» MICRO CONFIGURATION

You can tweak just about all parameters of *Micro* from within the tool itself, using the command bar. The `setlocal` command can be used to temporarily change a setting but you must use the `set` command to make permanent changes to the `settings.json` file.

One of the settings you'll want to enable is autosave. Run the `set autosave X` command to ensure that *Micro* automatically saves your file every X seconds. The default is 0, which disables the setting.

You can also save backups of the buffer. This ensures that if the terminal emulator window is closed without first saving the file in *Micro* or if *Micro* crashes, your changes aren't lost. If enabled, *Micro* stores backups of all open buffers in the `~/config/micro/backups` directory every eight seconds. The backups are purged as soon as you save the open buffer. To enable backups, run `set backup true`.

For a list of all configurable parameters, press `Ctrl+e` and then run the `help options` command. You can also view the current defined value for an option with the `show option` command. For instance, the `show autosave` command displays 0 by default, or the number you specified if you used to the `set autosave X` command.

Similarly, the `show colorscheme` command displays the colour scheme currently in use. You can run the `help colors` command to learn how to switch the colour scheme.

Hello, world!

The MIT licensed project is distributed as a batteries-included static binary with no dependencies. While some distros offer *Micro* in the software repositories, you can install it with the official script that installs *Micro* in your current working directory. You then have to move the executable into a directory in your `$PATH`. If you're uncomfortable running scripts, you can download the official tarballs, extract the files, then move the executable to a suitable directory:

```
$ curl https://getmic.ro | bash
```

```
<output snipped>
```

```
Micro has been downloaded to the current directory.
```

You can run it with:

```
./micro
```

```
$ sudo mv micro /usr/local/bin
```

You can now launch *Micro* with `micro`. To open an existing document, press `Ctrl+o`. The status bar at the bottom changes so that you can type in the path to the file you wish to open. *Micro* supports tab completion, so you don't have to type in the complete path.

One of the biggest pet peeves some users have with CLI utilities is how they are expected to learn new

keybindings for each one. However, most graphical utilities favour shortcuts such as Ctrl+s to save files, Ctrl+q to quit, Ctrl+o to open files and so on, and these and many more are the default on *Micro*, making it stand out from other command-line text editors.

Keybinding	Function
Ctrl+g F1	Open help
Ctrl+s F2	Save file
Ctrl+f F3	Find
Ctrl+n	Find next matching instance
Ctrl+p	Find previous matching instance
Ctrl+q F4	Quit <i>Micro</i>
Ctrl+c	Copy selected text
Ctrl+v	Paste selected text
Ctrl+x	Cut selected text
Ctrl+z	Undo
Ctrl+y	Redo
Ctrl+a	Select all
Ctrl+e	Open command bar
Alt+g	List common keybindings

For a list of all the defined keyboard shortcuts, press Ctrl+g or hit F1 to access the help. Note, however, that the function key shortcuts don't work on all terminal emulators. Press Ctrl+q to drop back to the document if you're in the help screen.

When you exit *Micro*, you're asked to confirm if you wish to save or discard changes. You might have to resize your terminal emulator horizontally if you can't read the complete message in the status bar.

Not all features are configured with a shortcut. For instance, there's no shortcut to Save As, or to search and replace words. These and various other operations can only be performed from the command bar.

Micro has robust mouse support, so you can also use a mouse click to position the cursor, double-click to select a word, and triple-click to select an entire line.

The command bar

Should you ever need to work with multiple files at the same time, you don't have to launch several instances of *Micro*. Instead, you can split the screen either horizontally or vertically. From the command bar, run the `hsplit <filename>` or `vsplit <filename>` to split the screen horizontally or vertically respectively.

You can also choose to run these commands without specifying a filename if you wish to open a blank document. You then have to press Ctrl+w to cycle through the different splits, or use the left mouse click to select the split you wish to work on.

To access the command bar, press Ctrl+e. You'll notice that the status bar at the bottom of *Micro*'s interface now features a '>' prompt.

For a list of all defined keyboard shortcuts, you can run the `help defaultkeys` command from the

command bar. This opens a pane in the bottom half of *Micro*'s interface that lists all the defined shortcuts split into different sections such as Navigation, Find Operations, Text Operations and others.

To save the current document with a different name, you have to first access the command bar by pressing Ctrl+e and then run the `save <path-filename>` command. Specify the complete path and filename, and *Micro* saves the file to the specified name.

The `replace 'text' 'replacement'` command replaces the first matching instance. To replace all instances, you need to run the `replace -a 'text' 'replacement'` command instead.

Run the `help commands` command for a complete list of commands on offer. Alternatively, once at the command bar, you can press Tab for an alphabetical list of available commands. Press Tab to cycle through the list of available commands until you find the command you're interested in. You can also type a letter and then press Tab, to view all available commands starting with the specified letter. For instance, if you type `h` and then press Tab, *Micro* lists `help hsplit` because only those two commands start with the letter `h`.

Micro features a built-in plugin manager that can be used to install plugins to improve *Micro*'s functionality. From the command bar, run the `plugin install plugin-name` command to install the plugin. *Micro* opens a separate output pane and informs you when the plugin is installed.

For instance, the command `plugin install monokai-dark` installs the monokai-dark colour scheme. You can then use the `set colorscheme monokai-dark` command to switch from the default colour scheme.

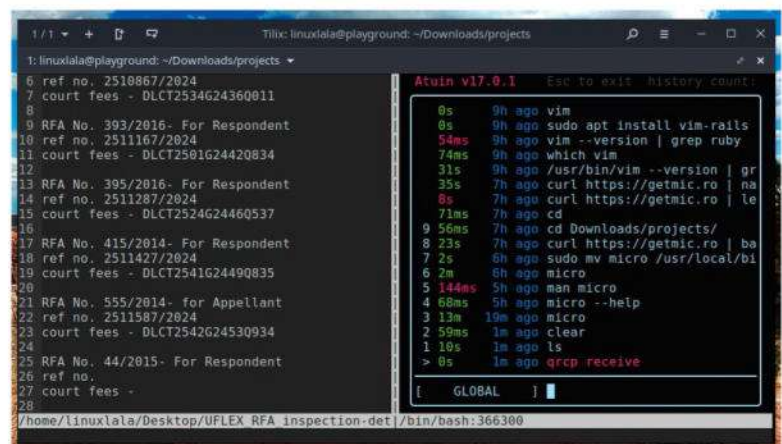
You can similarly install the word count plugin with the `plugin install wc` command and then use the `count` command, which displays the number of lines, words and characters in the open document in the status bar.

Refer to *Micro*'s website for more details about the available plugins, such as the command used to invoke them once installed. Each plugin has its own GitHub page with information about dependencies, if any.

If you've worked with *Nano* or other minimalist text editors in the past, you'll be pleased with the performance and features of *Micro*. **UX**

QUICK TIP

Tab completion works on the command bar, not just to specify filenames, but available commands as well. You can also use the up arrow key to cycle through all the previously executed commands from the command bar.



To launch an instance of the terminal within *Micro*, first split the screen and then run the `term` command.

» ENHANCE YOUR TERMINAL-FU Subscribe now at <http://bit.ly/LinuxFormat>

LINUX BASICS

Part Five!
Don't miss
next issue,
subscribe on
page 16!

Navigate your way around the filesystem

Nick Peers explains the basics of using the Linux filesystem, from managing your files to learning about permissions.



OUR
EXPERT

Nick Peers has spent enough time around computers to know the importance of a good file manager.

Files play a crucial role in any operating system, but they're especially important in Linux thanks to the way its ext4 filesystem is structured. In this tutorial, we'll take you through the art of navigating the Linux filesystem (FS) on your PC. We kick off with a quick tour of your desktop's file manager, based on the *Files* app found in the latest version of the Cinnamon desktop in Mint 22. We'll then delve into the key aspects of working with files. You'll get an overview of the ext4 FS itself, including how to mount additional drives so they're easily accessible.

We'll then take you on a quick tour of Linux's key directories and what you can expect to find inside, before tackling a key aspect of Linux files: permissions and file ownership. You'll discover how permissions are applied, what effect they have and how to change them. Get more from the *Files* app with the help of our walkthrough (page 54), which will help you with the day-to-day tasks of navigating, editing and managing files. And if you find all that a little dry, check out the



You can find out which file refers to which partition by examining the item's **Device** entry in the **Disks** utility.

box (page 55) on setting up a shared folder so you can easily transfer files over your network.

Not a Mint user? Not a problem – most of what we cover is easily translatable to your choice of distro.

Access Files

As you'd expect, access to *Files* is baked directly into the Cinnamon desktop – simply click the folder icon on

QUICK TIP

Expand the **Nemo** file manager's capabilities with plugins. Add rotate and resize options to the context menu when you right-click an image with `sudo apt install nemo-image-converter` or type `sudo apt install nemo-compare` to add options for comparing files.

» FILE PERMISSIONS EXPLAINED

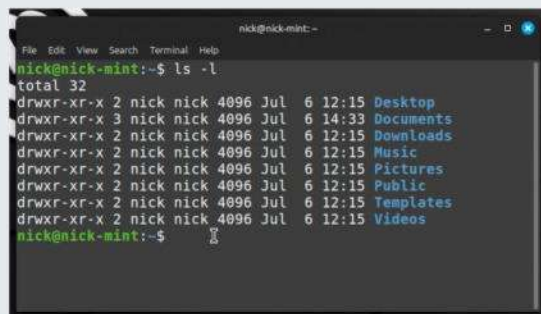
There are three levels of access to files and folders: read, write and execute, represented by the letters r, w and x in the terminal when you navigate to a folder and run the `ls -la` command. Look out for entries such as `rwX` (full access) or `r--` (read-only) next to files.

Not all file permissions reside directly with the file, however – the file's parent folder's permissions control certain functions. For example, if you want to view, create or delete a file, you need the parent

folder to possess both r and x permissions if you want to view it, and w if you want to delete it (or create a new file from scratch).

Complicating things further is the fact that there isn't a single set of permissions for files and folders; there are three levels depending on who's trying to access them (this is why you see three sets of `rwX` permissions next to files when displayed using the `ls -l` command).

The three levels apply to a file's owner (typically



The **'ls -a'** command gives you an at-a-glance view of the permissions of files and folders within a specific directory.

the person who created it), a specific user group (typically the same as the file's owner), and all other users. This allows files and folders to be

configured so different people have different access, enabling you to restrict access to read-only (or no access) for certain users or groups.

the taskbar to open a window that should be familiar to all PC users, including Mac OS and Windows switchers. *Files* takes the classic two-pane approach, with shortcuts on the left and a view of the selected folder, which defaults to your personal **home** folder. There are few surprises – the annotation (*right*) reveals the key elements of *Files*, and anyone familiar with the Windows File Explorer will find it's a simple transition.

You can easily access any part of your filesystem in *Files*, but its main use is for managing your own files in your personal **home** folder. Before exploring further, let's sidestep to examine the structure of Linux's FS.

Filesystem basics

Linux uses a different FS – ext4 – from other platforms, including Windows (NTFS) and the universal FAT32 and exFAT filesystems found on external drives. Before we dig too deep into ext4, it's important to understand that on one level it's very simple, particularly when viewed through *Files*. You'll see your personal **home** folder contains everything you need on a day-to-day basis, with familiar-looking folders like **Pictures** and **Documents** that are a staple of all operating systems.

Your **home** folder is also where many apps store your personal configuration settings, albeit in folders that are hidden from view by default. Any folder with a period (.) in front of its name, such as **.config**, is hidden by default in Linux. To make them visible in *Files*, open the View menu and tick Show Hidden Files, or press Ctrl+H if you're in a hurry. You'll see more folders and files appear, such as **.local**, **.cache** and – of course – **.config** itself. We recommend you only show hidden files and folders when you explicitly need to access them, so for now press Ctrl+H to hide them away again.

Delve deeper

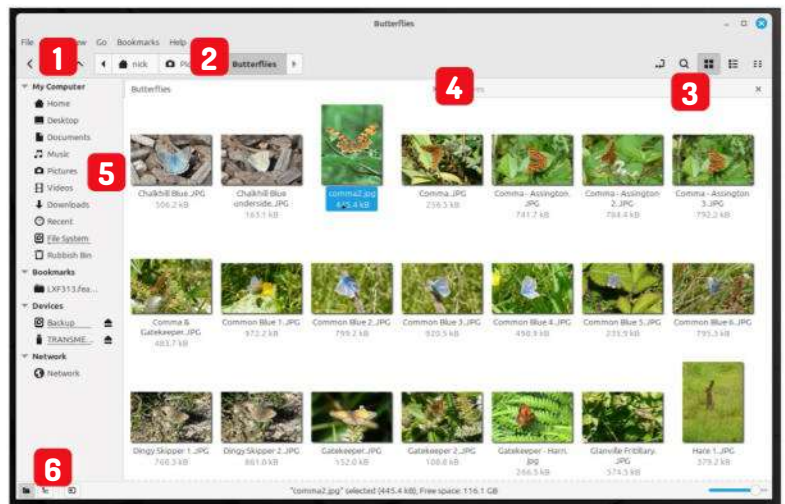
It's time to scratch the surface of ext4. One of its major differences compared to NTFS and FAT-based filesystems is that everything – including directories, other drives and even hardware devices – is referenced as files within the main FS.

There are three broad types of file: regular files, of course, and then directories, which are basically files containing a list of the files (and other information) inside them. Finally, we come to special files, which represent the physical devices attached to your PC, from drives and their partitions to printers and graphics cards. These reside within a dedicated directory (**/dev**).

Drives and partitions are represented by files beginning **sd** – **sda**, **sdb** and so on for physical drives, and **sda1**, **sda2**, **sdb1** and so on for the partitions (or volumes) within each drive. To find out which files represent each drive and partition, open the *Disks* utility (it's under Preferences on Mint's Start menu) and select your hard drive from the left-hand panel.

You'll see your drive shown as a bar chart, displaying all its partitions (a minimum of three). Your boot partition – the one on the left – is selected by default, and if you look at its Device reference, it says **/dev/sda1**. This reveals that the boot partition is represented in Linux by the file **sda1**, stored inside the **dev** directory. Select the other two partitions and you'll see they're marked **sda2** and **sda3** respectively. If you have a second hard drive or USB flash drive attached, select that and you'll see partitions are labelled **sdb1**, **sdb2**

NAVIGATING THE FILES APP



- 1 Navigation tools**
Use the left and right arrows to switch between previously visited and next locations. Click the up arrow to go up one level.
- 2 Breadcrumbs**
Use the breadcrumbs view to move more quickly between folders in the current path – click a folder to jump to it.
- 3 Search**
Click this button to access the Files search tool, which can search by filename or content.
- 4 Tabbed browsing**
Like all good file managers, Files enables you to browse to multiple locations within one window using tabs.
- 5 Navigation pane**
Quickly jump to popular locations, such as your personal home folders, plus access external drives and more.
- 6 Navigation pane options**
Click to switch between the Places view (default) and Treeview (folder hierarchy), or hide the sidebar completely.

and so on. If you have a DVD drive attached, this is allocated **/dev/sr0**.

These files can't be accessed directly, so to make the drives accessible, Linux mounts them as different folders within its filesystem. For example, your main system partition is mounted to the top-level – or root – directory, which is represented as **/** in Linux file paths (so **/dev** is a folder found at the root level).

When it comes to your personal **home** folder, this is stored inside another top-level folder (**/home**) – note, on some Linux distros (but not Ubuntu or Mint), the **home** folder is stored on its own dedicated partition. This reveals how confusing the ext4 FS can become – drives can be mounted anywhere within the FS, so while you might think **/home** is part of your boot partition, it isn't always the case. Nevertheless, it's logical once you realise the filesystem sits at the top of the hierarchy, with physical devices slotting into obvious places where you can always find them.

Accessing external drives

Other internal and external drives aren't usually mounted as folders by default, but the good news is they're still easily accessible in Linux, even those using different filesystems, such as NTFS or FAT. In the past, you had to install dedicated drivers to access these, but now read/write support is built into Ubuntu and its derivatives, like Mint. Simply connect the drive and it'll open a new *Files* window, indicating it's been mounted.

You'll also see a new entry referring to the drive appear under Devices in the navigation pane on the left, enabling you to browse and access its contents

QUICK TIP

To make an internal drive mount automatically when Mint starts, open the *Disks* utility. Select your drive, click the settings cog icon underneath the bar chart, then choose Edit Mount Options. Simply click the User Session Defaults switch off and ensure Mount At System Startup is ticked. Click OK.



easily – you'll also see an eject button you should click to unmount the drive before physically removing it. If you're wondering where external drives are mounted inside the FS, it's inside the **/media/user** folder.

It's worth noting that while external drives such as USB flash drives are mounted automatically when your PC starts, other drives – including internal hard drives – aren't. They appear in the Devices menu, but the lack of an eject button icon next to them indicates they're offline. This isn't necessarily a problem – simply click the device and it mounts, but what if you want the drive mounted automatically so its contents are always available, not just to you but to apps and other parts of your system, too? The answer lies in the Quick Tip (*previous page*).

Take a tour

Although it can look daunting at first glance, Linux's filesystem is quite elegant and straightforward to navigate. If you type `cd /` followed by `ls -l` into a terminal window – or navigate up two levels in the *Files*

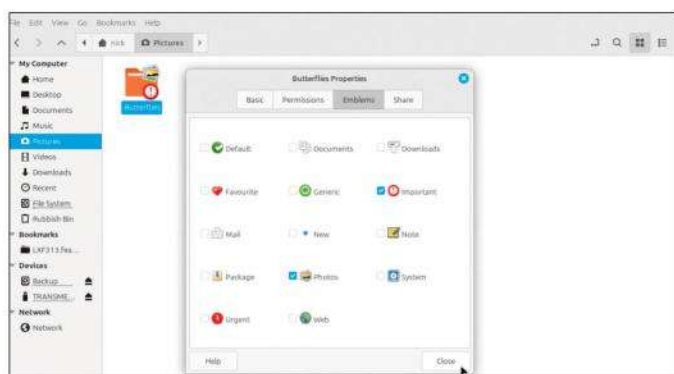
app by clicking the upward arrow button twice – you'll find yourself at the root directory (**/**) of your filesystem, with a directory list that's consistent across all distros.

Some – such as **/boot** for the boot files – are self-explanatory, while you'll also recognise the **/dev** (devices) and **/home** folders from our earlier explanations. But keep an eye out for the following:

- **/bin** – this directory contains several key utilities that you'll use regularly in the terminal, including `ls` for listing the contents of directories and `cp` for copying files.
- **/etc** – contains many configuration files, such as **/etc/samba/smb.conf** for editing network shares.
- **/media** and **/mnt** – contain mount points for removable drives (**/media**) and internal drives (**/mnt**).
- **/root** – the root user's home directory.
- **/usr** – home to non-system-critical programs, libraries and other shared resources.
- **/var** – home to files that often change, like emails.

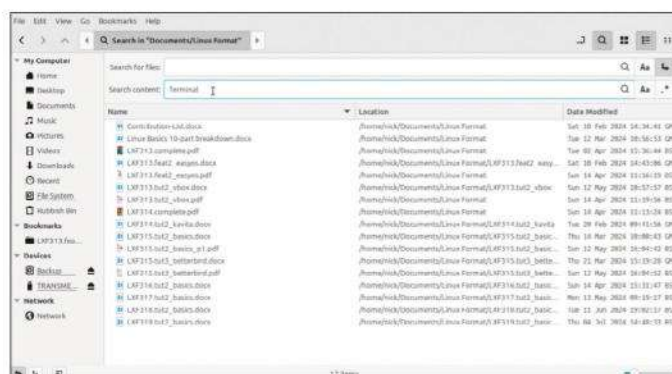
Sometimes you may find yourself having to edit files in these directories – the best way to do this is through

DO MORE WITH MINT'S FILE MANAGER



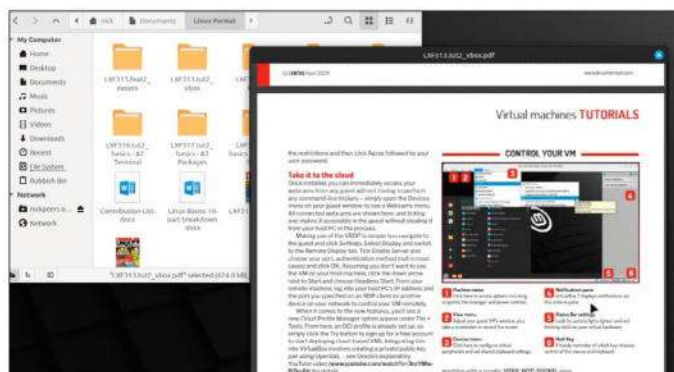
1 Customise folders

Right-click a folder to change its colour or choose Properties > Emblems to add up to four badges to the folder to help identify it. There are 14 to choose from, covering both document types as well as statuses such as Default or Urgent. Tick up to four boxes – the emblems appear instantly – before clicking Close to finish.



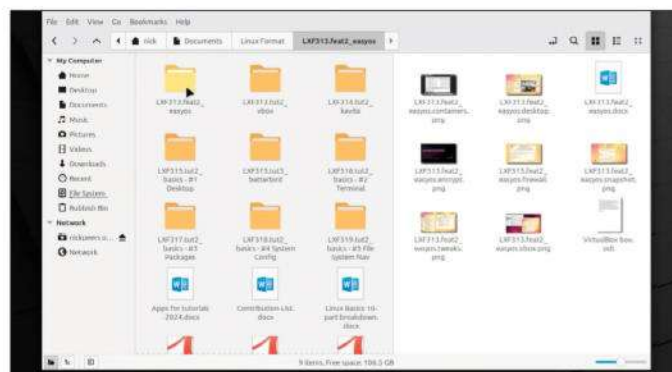
2 Find files

Click the magnifying glass icon to search the current folder. You can search for specific files or perform a keyword-related search based on the file's content. Enter your keywords and press Enter – results appear in a list below the search pane. Use the buttons on the right to make the search case-sensitive, recursive (search subfolders) or include regular expressions.



3 Take a peek

Mint's *Files* utility comes with a preview tool, enabling you to view photos and other supported file types (including office documents and PDF files) without opening them. Simply select a file in *Files* (including from search results) and press Space to pop up a preview. Press Space again to close the file.



4 Switch to dual-pane view

If you struggle to move files between tabs, but don't want to open separate windows to do so, choose View > Extra Pane (or press F3) to divide the current window in two. Click on a pane to select it, then navigate in the usual way. You can then easily copy and move items between two locations using drag and drop.

the terminal: `cd` to the directory, then use a text editor such as *nano* to edit the file. In most cases, you need to prefix the command with `sudo` to edit the file as the root user.

File permissions

A key part of the ext4 filesystem is security. This is applied to files (including drives and special files) through permissions, which limit access to and use of files based on who's trying to use them. The box (page 52) reveals what these permissions are and how they're set at three different levels depending on who's trying to access them. By way of example, your personal **home** folder, as well as any drives mounted in the **/media/user** folder, are owned by your username and user group (which has the same name as your username).

When you copy files into a folder, they keep their existing permissions, but become subject to that folder's permissions. *Files* also offers some visual clues as to their permissions by displaying a lock symbol on files or folders that your user doesn't have full access to – they may be read-only, or you may be locked out.

Permissions – and file owners – can easily be changed, both from within the terminal, as we revealed in **LXF316's** *Linux Basics* tutorial, and in *Files* itself. However, don't just rush to make changes – you could easily lock yourself out of your entire Linux installation if you make the wrong change to key files.

To avoid this, only change permissions on files within your own **home** folder or on external drives mounted through the **/media** or **/mnt** folders. Leave all other permissions and owners alone.

Change file and folder permissions

The most common reason for changing permissions comes when you download a program from the internet, such as a DEB installer or Applmage portable

application. These files can only be run as programs if they have the executable (x) permission set, which must be done manually for security reasons. After downloading the file from a reputable location, right-click it and choose Properties. Switch to the Permissions tab and tick the Allow Executing File As Program box before clicking Close.

When it comes to altering permissions on other files, you need to be the file's owner – or the root user – to make changes. Thankfully, the *Files* utility makes it easy to obtain the required level of access to do so: simply right-click inside the current *Files* window and choose Open As Root. Enter your password and a new *Files* window appears with a tell-tale red Elevated Privileges banner, indicating that any actions performed inside this specific window will be done as the root user.

Ownership and permission changes can be applied to a single file or folder, or you can select multiple items at once by holding the Ctrl key as you click each one. Once selected, right-click the item(s) in question and choose Properties > Permissions. Here you'll see a series of drop-down menus enabling you to easily change ownership and permissions. If switching ownership, select your chosen owner and/or group from the relevant drop-down menus. You'll see lots of unfamiliar entries – these are system-generated users and groups designed to do specific jobs for security reasons, so leave these alone and focus on those you recognise, such as your own username.

When editing file permissions, use the Access drop-down to set permissions: None, Read-Only, or Read And Write. Folders show two access drop-downs: Folder Access applies to the folder itself, while File Access enables you to change the permissions on any files inside the folder, too – simply select your desired permissions across the three drop-down menus and click Apply Permissions To Enclosed Files. **LXF**

QUICK TIP

Choose **Edit > Preferences** to open a multi-tabbed set of options. Go through each section seeing what you can customise – for instance, tick **Show Advanced Permissions** In The **File Property Dialog** under **Display** to change the way permissions options are displayed.

» SHARE FOLDERS

Although it's technically possible to configure sharing using the *Files* utility, experience has taught us that it's not always the best way to do so. Instead, we recommend using the terminal. First, install two packages: **samba** for sharing with other platforms, and **wsdd** to make your computer visible to Windows hosts:

```
$ sudo apt install samba wsdd
```

Next, add a network password for your user account, which you'll need to log on to your shared folder from elsewhere – substitute 'username' with your actual

username, such as 'nick':

```
$ sudo smbpasswd -a username
```

Next, you need to create your shared folder:

```
$ mkdir share
```

Now we have to edit a configuration file to turn the folder into a shared one and set the correct permissions for it:

```
$ sudo nano /etc/samba/smb.conf
```

Go to the end of the document and enter the information you can see in the screenshot (above-right) from [Mint Share] downwards. You need to adapt this to your



Configuring shared folders via the **smb.conf** file is our recommended way of setting up Samba shares in Mint – **Files** is too unreliable.

own needs – change the path to point to your user folder, and you can make the folder read-only or accessible by guests if you wish.

Once done, press Ctrl+X, press Y, then restart Samba:

```
$ sudo service smbd restart
```

You can now access and use your shared folder.

» **IMPROVE YOUR LINUX SKILLS** Subscribe now at <http://bit.ly/LinuxFormat>

Geek out with NixOS

Mats Tage Axelsson shows you how easy it is to use NixOS and the challenges of installing new applications. All 100,000 packages.



**OUR
EXPERT**

Mats Tage Axelsson is still searching for that elusive ideal way to run his computer and reach OS nirvana. Changing all aspects of his system, he may soon succeed.

Distributions pop up like mushrooms in autumn, with most being new spins of the main ones: Debian, Ubuntu and Red Hat. The major differences are the package manager, choice of desktop and default software. Maintainers aim new spins at specific users – Kali Linux for security professionals springs to mind.

These distros all deliver binary programs, a practice that means they are not optimised. Instead, they work on all the platforms you choose. Others have a recipe approach, where the system describes how to compile your package for the current system. Gentoo and Arch use this approach. They both require effort to install; well worth it, according to their most fervent users.

You can spend your entire life arguing which approach is better and get nowhere. It all depends on your needs. The same goes for NixOS. All they provide are recipes to compile your packages and the system compiles the package on install by default. However, most users will probably use the cache stores, where it can pick a suitable binary. The big difference for NixOS is that you have atomic upgrades and rollbacks. In short, you can go back to where you were with a reboot and choose an older generation.

Choosing a distribution comes down to what you are doing. Nowadays, casual users can pick up Ubuntu and stick with it. The reason is obvious: it is well known and has simple tools to help you choose your apps. It does have one quirk, though: while it uses the Debian APT system for package management, it also defaults to using Snaps for many parts of the system and you can also choose a Flatpak or an AppImage. This isn't necessarily bad, but it is a mess.

What is NixOS?

In short, NixOS uses the *Nix Package Manager* to handle packages. The most interesting part is not that

```
matstage@matstage:~$ nix-env --list-generations
23 2024-05-29 10:09:12
24 2024-06-18 16:11:11
25 2024-06-18 16:11:13
26 2024-06-18 17:07:48 (current)
matstage@matstage:~$ nix-env --delete-generations 18d
matstage@matstage:~$ nix-env --delete-generations 23
removing profile version 23
matstage@matstage:~$ nix-env --list-generations
24 2024-06-18 16:11:11
25 2024-06-18 16:11:13
26 2024-06-18 17:07:48 (current)
matstage@matstage:~$
```

Once you use the *Nix Package Manager*, you have to learn how to handle generations to help you save disk space.

```
matstage@matstage:~$ nix run nixpkgs#dash
$
matstage@matstage:~$ nix run nixpkgs#fish
Welcome to fish, the friendly interactive shell
Type help for instructions on how to use fish
matstage@matstage ~ (master)> ls -l ~/AppImages/
total 493520
-rwxr-xr-x 1 matstage matstage 208573354 Jun 6 20:59 gearlever_anymtype_0b5aba.
ppimage*
-rwxr-xr-x 1 matstage matstage 101703502 May 18 11:39 gearlever_stomickwallet_0b
ppimage*
-rwxr-xr-x 1 matstage matstage 151176384 Dec 30 00:05 gearlever_nextclouddeskto
client_4a3ab2.uppimage*
-rwxr-xr-x 1 matstage matstage 43881664 Jun 2 18:32 gearlever_warpp_bc190f.app
matstage@matstage ~ (master)>
```

With the experimental feature 'commands', you can run any shell with a single command, compiled and installed on the fly.

it uses its own scripting language to achieve this. What is fascinating is that when installed, all binaries sit in a store, separate from the structure defined by the Linux Standard Base. Your */usr*, */var*, */opt* and so on.

NixOS instead links any binaries out of the store into the directories required. It is so clever that you can have several versions of the same applications and even libraries installed in parallel. This takes care of dependency hell but comes with its own challenges – namely, each application might install the same package as another. And you end up with a lot of copies of binaries – bit-identical copies.

The solution to this is simple: using a hash comparison algorithm, *Nix*, the package manager, can find the duplicates, relink and remove ones without a link. Garbage collection, they call it. Why the focus on dependencies? Well, the target group is developers. They tend to end up having small differences on their machine between them, creating difficult-to-solve problems for new members of the team.

Using the approach in *Nix*, you can define the entire environment without changing what you are already using. You can compare this to virtual environments in Python, except it is completely independent of the programming language used. On the subject of languages, *Nix* is also a programming language. More exactly, it is a declarative expression language, in line with the main philosophy of NixOS: reproducibility. The idea here is that you should be able to write a declaration in the *Nix* language that you can apply again and again, even on different computers. All while ending up with exactly the same system each time.

One configuration file!

In ordinary distributions, you install what they have and must pick each time. The system ends up working in a similar way to the old system but not exactly the same. This causes problems for developers and potentially strange bugs for users. For regular users, you can use a

QUICK TIP

Try *Nix Package Manager* first – you can create all the files needed for a fully-fledged NixOS system to rock your world. The files hardly differ at all when you eventually migrate.

single file to define all the software you want to use and install on many computers.

So far, this file has been **configuration.nix** but things are happening fast, so get ready for *Flake*.

How do you install it? This question soon becomes two questions: how do you install the system and how do you install applications?

To answer the first question, you use the *Calamares* installer (<https://calamares.io>). This is what the main install ISO has when you pick the Graphical ISO image from <https://nixos.org/download/>. If you have previously installed any of the main distributions, you will recognise this interface. It is a simple step-by-step procedure to choose things that you want to set for your system, the details being time zone, locale and so on. You end up with a simple desktop with a limited number of applications.

That method is the new default install option, introduced in the latest versions. You can still choose to use the command-line option, where you partition and mount your disks manually, generate and edit the configuration, and install with a final command. Described that way, it may seem simple but you have to be confident with the command line and what partitioning scheme to choose and so on. You do have the choice of adding all applications you need and using *Home Manager* at the first instance.

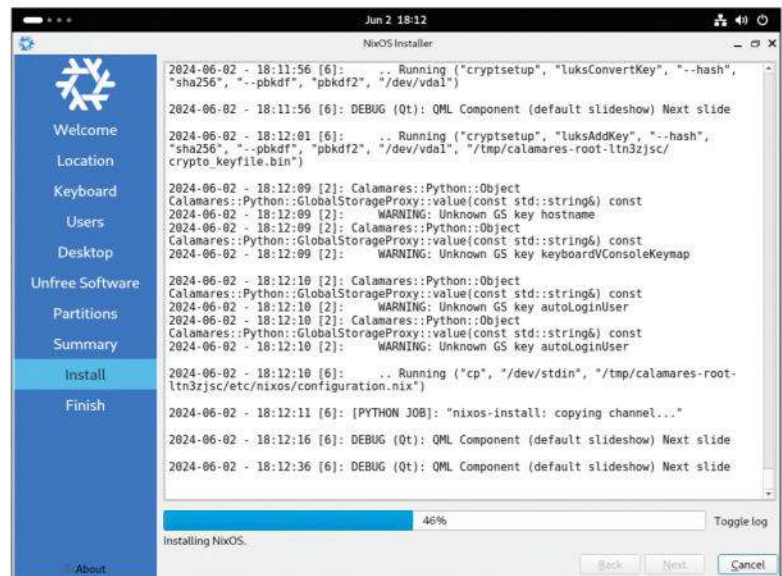
For more advanced users, you can choose to create the new NixOS system in a chroot and install it on your currently running distribution.

Surprisingly, the *Calamares* installer does not include a way to fetch your configuration from disk. Why not from your cloud storage? That would be a worthy project to make NixOS even better.

Once you have the system installed with your preferred desktop, you need applications. While you can set *ApplImages* and *Flatpaks* to run, the main way to add applications is through the **configuration.nix** file. The way you do this seems fiddly at first, but is actually simple. You open **configuration.nix** and add the name of the application in the file:

```
environment.systemPackages = with pkgs; [
  wget
  vim
  firefox
];
```

As you can see, it says **SystemPackages** – that means you are adding applications to the general



system. In contrast, you may want to add software only to your own user. There are several ways you can add and remove software for your own user: *profiles* is one and *Home Manager* is another. This is important when you start using NixOS in more advanced ways.

Sometimes, software goes under slightly different names depending on the distro. To make sure you get the correct package, check with the NixOS search site (<http://search.nixos.org>). When you search for *Firefox*, you get a long list of choices, including side projects – choose the shortest name, unless you know better.

Once you have saved the file, you need to build the system and switch it in with the following command:

```
$ nixos-rebuild switch
```

The build takes some time, so make sure you change all the packages you want to have on your system. What happens when this command runs is that it compiles all the software, or fetches it from cache, then creates and links a new generation. In the long run, your disk fills up with generations. To address this, there is a command to remove old generations:

```
$ nix-env --list-generations
```

```
$ nix-env --delete-generations 30d
```

The second command is wise to put in as a service that runs on a daily or weekly basis. Hopefully, you never want to roll back further than a month. You can also choose how many generations to keep if you wish.

During install, you will run into the now infamous 46% lock, which is when the system compiles and links packages.

QUICK TIP

The *Home Manager* is a powerful tool for setting up your personal system to suit your own taste. With this, you can create a super-slim base system and expand it for your own user. Other users on the same system can keep things slim.

» GUI INSTALLER

Using NixOS requires you to be handy with the command line and editing files. This limits it to fairly advanced users. Any newbies will have to ask for help adding new apps. Fortunately, adding an app only entails adding its name to the relevant config file. If you need to change settings, you have the challenge of learning the Nix language or stealing other people's configurations.

Projects that tried to handle these problems include *nix-gui* and *nixos-*

manager. They have a simple list view, where you can choose packages and create settings. Unfortunately, these projects have not had updates for four years. Despite that, they are interesting starting points for an easier way to set up your NixOS system: <https://github.com/ahoneybun/nixos-manager> and <https://github.com/nix-gui/nix-gui>.

Another interesting project is *nix42b*, which delivers some convenience for choosing and adding your packages, and

adds the option to run, install and tweak packages. *Nix42b* is not yet ready for primetime, but it gives us a view into how it can be done: <https://gitlab.com/julien-dehos/nix42b>.

After a taste of a graphical installer, you can see what may be on the horizon. Features lacking in these packages are manually adding channels and an IDE of sorts to import and change settings to your own liking. Connect all that to the *Flake* functions and you'd have a winner!



In your configuration file, you can set which programs are available for a specific user. Similar to the Home Manager.

```
#jack.enable = true;

# use the example session manager (no others are packaged yet so this is
# no need to redefine it in your config for now)
#media-session.enable = true;
});

# Enable touchpad support (enabled default in most desktopManager).
# services.xserver.libinput.enable = true;

# Define a user account. Don't forget to set a password with 'passwd'.
users.users.matstage = {
  isNormalUser = true;
  description = "Mats Tøge";
  extraGroups = [ "networkmanager" "wheel" ];
  packages = with pkgs; [
```

of using *Nix* is the experimental feature `nix-command`.

This lets you choose a package, run it and then stop. In this case, *Nix* creates a temporary environment where this package runs and starts it. When you are done, the files are waiting for you for the next time you need it unless you clear it with your garbage collector. Try the Fish shell:

```
$ nix run nixpkgs#fish
```

The first time, it downloads and compiles or fetches everything,

making it a bit slow, but on subsequent starts it is instant. Once you are happy that this is the shell for you, you can make it permanent using either *Home Manager* or adding it to your configuration file.

QUICK TIP

Using **Flatpak**, **AppImages** work with a single configuration line while **Snaps** are possible, but you need to put more effort into doing it. You can find a flake for this here: <https://flakehub.com/flake/io12/nix-snapd>

From this default install, you end up following the NixOS 24.05 channel. A channel is the same as a repository; it defines what version you are using. For some projects, you add another channel that has other software not delivered by the main distribution.

The strongest argument for NixOS is the package manager. The ease with which you can try a new shell or application is fabulous. Once you have *Nix* on your 'foreign distribution', you can take advantage of these capabilities. Now, comes the same question again: how do you install it?

Well, you do have a multitude of choices, one of which is running all commands by hand. The most straightforward is the script at <https://nixos.org/download/>. On the page, it discusses whether to use the multi-user installation or not. Unless you do not have root, use multi-user.

One alternative is the Determinate Systems installer (<https://determinate.systems/posts/determinate-nix-installer/>). This boasts better control of the install, enabling you to test it out and offering an uninstaller. If things go wrong, this is a godsend, but you may also end up with a `nix-store` that is too large.

These scripts do almost the same thing, barring the uninstall option added by the second choice. Here is a short summary of what they do to your system:

- Create a directory in root `/nix`.
- Create `nixbld*` users.
- Install the `nix` binaries.
- Set profiles up.

All these things are necessary for driving the features of the package manager. You can find links in your `home` directory to the software you have added and the ones you have used. One of the brilliant parts

Home Manager

Since you have more than one computer, you can create your own `home` directory. In the past, advanced users had dotfiles backed up using a git. Wait a minute, that is still common, and you might use it. If you do, consider using the *Nix Home Manager* – it can handle all your dotfiles. In addition, you can set all the applications you need in the same file or directory.

Home Manager requires flakes for operation. A flake is a directory with a file in it that guides the install using the *Nix* language. You must have *Git* installed for this to work. To set up the *Home Manager* flake, you add a channel to your configuration.

There are three ways to do this, but when you install it on a foreign distribution, only one is available. After having *Nix* installed, you can add it through a few steps. First, you add a channel to follow – it aims to follow the unstable branch of *Nix*:

```
$ nix-channel --add https://github.com/nix-community/home-manager/archive/release-24.05.tar.gz home-manager
```

As eagle-eyed readers can see, the first parameter is a packed file. This has descriptions for the channel, the main being `default.nix`. Now that you have the channel defined, you must update your local system:

```
$ nix-channel --update
```

This action makes the `home-manager` package available. Next you install the actual *Home Manager*:

```
$ nix-shell 'home-manager' -A install
```

Using *Home Manager*, you can add applications and handle your dotfiles. To start with, turn off the possibility to change and erase all your configuration files. To do this, you need to add the following line to your shell configuration:

```
$HOME/.nix-profile/etc/profile.d/hm-session-vars.sh
```

Get used to *Home Manager* before you start experimenting with handling your own dotfiles – you could clobber your settings when changing over.

Your first application

The safest and recommended way to install software is to edit your `~/.config/home-manager/home.nix` file. This script creates a file with minimal settings, packed with comments. Read the comments to develop your configuration. In this file, you see only the `ddgr` and `yt-dlp` packages:

```
behavior. It is therefore highly recommended to use a release of Home
Manager that corresponds with your chosen release of Nixpkgs.

If you insist then you can disable this warning by adding

  home.enableNixpkgsReleaseCheck = false;

to your configuration.

this derivation will be built:
  /nix/store/fzdli3cav7gls7w02h5ka5sd343lxjaj-hm-news.json.drv
building '/nix/store/fzdli3cav7gls7w02h5ka5sd343lxjaj-hm-news.json.drv'...

There are 159 unread and relevant news items.
Read them by running the command "home-manager news".

All done! The home-manager tool should now be installed and you can edit

  /home/matstage/.config/home-manager/home.nix

to configure Home Manager. Run 'man home-configuration.nix' to
see all available options.
matstage@matstage:~$ 3-
```



```
{ config, pkgs, lib, ... }:
{
  home.username = "matstage";
  home.homeDirectory = "/home/matstage";
  home.stateVersion = "23.11"; # Please read the
  comment before changing.
  # The home.packages option allows you to install Nix
  packages into your
  # environment.
  home.packages = [
    pkgs.ddgr
    pkgs.yt-dlp
  ];
  # Let Home Manager install and manage itself.
  programs.home-manager.enable = true;
}
```

The format works like namespaces. When you have values for the home, you put it behind a dot with packages under in a tree structure. The square bracket notation saves you writing `home.packages` repeatedly. You have many more details to learn for configuring packages in the Nix language.

The big difference

As you can see, installing an application is cumbersome compared to other distros. Even if you don't install new apps, it can stop the adoption rate for many people.

You may wonder what the point is when it is more complicated to install new software. The main benefit is that you can both run any new software with a single command and create control files using flakes that run in isolation from your system.

The isolated nature of running software is particularly useful when you need to try new software or want to try a new configuration. With other systems, you would have to jump through hoops to find an isolation strategy; with Nix you get it built in.

When you are trying it out, you should write Nix files, which enables you to keep the file, erase everything else and create the same thing elsewhere.

No solution ticks all the boxes. NixOS definitely has its weaknesses. A serious one is that there is no application store where you can pick up applications. There are efforts underway to handle this drawback but they are far from production-ready nor actively maintained (<https://github.com/ahoneybun/nixos-manager> and <https://github.com/nix-gui/nix-gui>).

Another drawback is that the Nix language, which you must learn, is simple to start with but deceptively hard to use in real life. Most users copy a configuration from someone else and tweak it. Most do not learn Nix. There are also documentation shortcomings.

Using Scheme, the way Guix has done, would have been a better solution. Adding some libraries to any

» NOTHING FLAKY HERE

A flake is a way to define specific packages or environments. You point to existing packages using inputs. Then you define what you want to use – an output. The input is usually another Nix package but you can also use an arbitrary directory, say a source tree. When you do that, you need to define in output how to compile the source tree. Naturally, GitHub support is excellent but any file will do, as long as you can unpack it and define how to handle it.

An example is when you want to compile a package, you define the `stdenv.mkDerivation` with a name, the source and the two phases: `buildPhase` and `installPhase`. This way, you set the command `gcc -o ...` to compile and `mkdir -p ...` to install in the two phases. The compile command can be any compiler. NixOS supports most compilers but if you need one that is not yet supported, you add it in your input.

You can create your entire development environment with this one file and have it look the same on all systems on which it's run. On top of that, it doesn't change the system you're using. This construct can also copy files to the filesystem; imagine you want to put add unique fonts – make a flake that copies the files to the correct place.

The functions to do this exist but NixOS makes it trivial to have it fully insulated from the main system.

existing language would have been better than coming up with a new one.

If the concept seems wonderful but you want other ways to install applications, you can use `ApplImages` and `Flatpaks` to pick up software. In some cases, you can only find your favourite software that way. With 100,000 packages available, though, this will be rare but it is still possible.

There is even a flake to support Snaps on NixOS, you can find the project on `FlakeHub`. This project creates a Snap daemon that you can use to install Snaps on a NixOS system: <https://flakehub.com/flake/io12/nix-snapd/>.

In good Nix

NixOS is a fascinating project that is stable enough to be your production environment. It also has the largest repository, with over 100,000 packages, making it the unbeaten champion. When you use NixOS, you can easily change things, with the confidence of a simple rollback if things go wrong.

To use NixOS, you do need to have some confidence in using the command line and the willingness to pick up some understanding of the new language. The latter becomes essential if you want to tweak your programs yourself, although you can often cut and paste whatever configuration you already have into a Nix language snippet. [LXF](#)



■ If you have tried a new version of NixOS and want to roll back, choose the old configuration at boot.

» FIND MORE WAYS TO USE LINUX Subscribe now at <http://bit.ly/LinuxFormat>

BACK ISSUES » MISSED ONE?

ISSUE 318 Summer 2024

Product code:
LXFDB0318



In the magazine

Find out how to boost your VPN privacy and set up a free VPN, and continue keeping your secrets safe with our *Roundup* of security-conscious chat services. Plus, try our tutorials on everything from tweaking your install settings to ray tracing on your Steam Deck, and plenty in between. And don't forget to read our packed news, reviews and Raspberry Pi sections to stay up to date with open source.

ISSUE 317 August 2024

Product code:
LXFDB0317



In the magazine

Don't consign your old machines to the scrap heap – they can still be put to work with Linux. And join us as we dive deeper into Linux to explore virtualisation, containerisation and virtual filesystems. We also have tutorials on managing software and files, emulating VDUs, creating old-school pixel art, and more. Plus we have our usual news, views and in-depth reviews.

ISSUE 316 July 2024

Product code:
LXFDB0316



In the magazine

Transform your desktop with hot KDE Plasma, and join us as we start our journey inside Linux to discover how it works. Plus, choose the best text editor for your writing projects, find out about the phone firm making an environmental impact, build your own weather machine, get back to basics with the Linux terminal, keep your video calls private, and enjoy a wealth of reviews, news and views.

ISSUE 315 June 2024

Product code:
LXFDB0315



In the magazine

Discover how to upgrade to Ubuntu 24.04 LTS and join us on a journey through the world's most popular distro's 20-year history. And don't miss our *Roundup* of hacker distros, an in-depth look at the impending Epochalypse, detailed tutorials on live-streaming audio via the terminal, setting up Gnome, editing images in *Krita*, and archiving C64 tapes, and plenty more besides.

ISSUE 314 May 2024

Product code:
LXFDB0314



In the magazine

Join us as we explore Linux's infiltration of Windows, and discover how you can also get Windows apps running in Linux as we uncork the latest *Wine* release. And let's not forget a *Roundup* of time trackers, tutorials on vulnerability audits, setting up a home ebook server, streaming games, and more, along with distro reviews, news, and packed Pi and *Adminsteria* sections.

ISSUE 313 April 2024

Product code:
LXFDB0313



In the magazine

Discover how to use the ultimate hacker's toolkit, staying out of trouble while doing so. And join us as we take the Puppy Linux developer's new distro for a run and explore its container features. Plus, we have a *Roundup* of retro-gaming distros, an in-depth look at filesystems, tutorials on adding plug-ins to our *LXF* shell and getting more from your VMs, plus news, reviews and oodles more.

To order, visit www.magazinesdirect.com

Select **Single Issues** from the tab menu, then select **Linux Format**.

Or call the back issues hotline on **0330 333 1113**
or **+44 (0)330 333 1113** for overseas orders.

Quote the product code shown above and have your credit or debit card details ready.

USA? EU? THE MOON?

UK readers
turn to
p16

SUBSCRIBE!

Don't wait for the latest issue to reach your local store – subscribe today and let *Linux Format* fly straight to you.



2 GREAT WAYS TO SUBSCRIBE

Print + NEW digital access or digital-only!

LINUX FORMAT
The #1 open source mag

BOOST YOUR VPN PRIVACY
Tweak and hack your network settings for snoop-free browsing and downloading

GET A FREE VPN! p32

PLUS: HOW TO

- Code classic Amiga-style demos
- Edit and control PDFs without Adobe
- Create a modern note tool in Fyne

DIY PI ROUTER
Build your own custom Pi-based Wi-Fi router

RAY TRACING
Enhance your Steam Deck gaming graphics

MS OFFICE RESCUE
Save old Office docs that Microsoft can't

THE PI GETS AI
First look at the official AI accelerator

MOD UBUNTU
Enhance your install with secret settings

PODCAST IT!
The apps, mics and tips for a pro podcast

SECURE YOUR CHAT
The message service that really protects your privacy

» USA
From \$135.49
For 13 issues

» REST OF THE WORLD
From \$135.49
For 13 issues

» EUROPE
From €119.49
For 13 issues

IT'S EASY TO SUBSCRIBE!

Visit www.magazinesdirect.com/linux-format

Call +44 0330 333 1113

Lines open Monday-Friday, 9am-5pm GMT

Or 1-844-779-2822

Lines open Monday-Friday 8.30am-5pm EST

*We don't actually deliver to the Moon. Yet.

RAID

Expand your storage with high-speed SSDs

Although he likes to think he lives life in the fast lane, **Neil Mohr** isn't sure what to do with 6TB of 3GB/s storage...



**OUR
EXPERT**

Neil Mohr is so old he remembers thinking a 16KB expansion pack was a bit over the top.

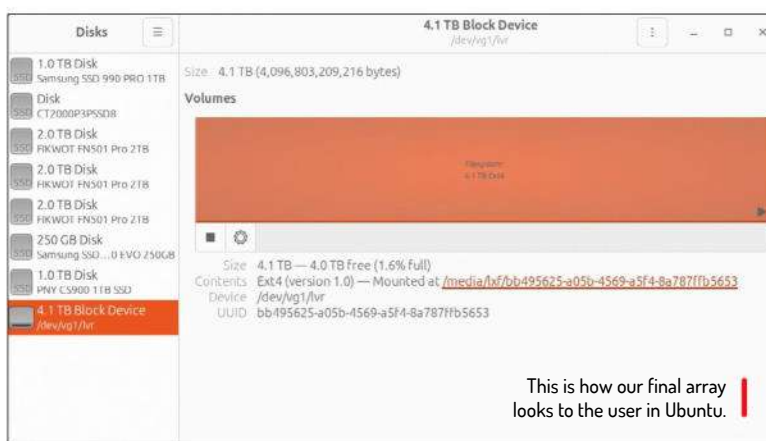
How do you increase storage on a system with no more drive slots? That's a reasonable question, especially when most modest motherboards only offer two NVMe slots.

As storage prices have spiked over the last year, it's not too hard to believe you have a few older NVMe's hanging around, or if you have a system with a smaller NVMe boot drive, you're looking at options to expand your main storage pool.

Sure, there are older SATA options, but why would you opt for 500MB/s speeds when even PCIe v3 NVMe offers 2GB/s? While speed is a consideration here, the real reason for this tutorial is about adding NVMe slots to a system – while 4+ SATA ports is the norm, NVMe numbers remain limited.

As NVMe is effectively bolted directly to the PCIe bus, add-in PCIe adaptor cards are a relatively straightforward option, supporting speeds all the way to PCIe v5. See the boxout (right) for possible options (also the review on page 19) – these enable you to add anything from one NVMe drive to multiple drives, depending on your budget.

What we're going to do here is take a four-slot M.2 NVMe expansion card and use it to add three 2TB NVMe drives to an existing Ubuntu system. Previously, we would have used *mdadm* to create a raw RAID 5, but as we're in modern times, we'll be using LVM and its built-in RAID facilities. This makes managing filesystems (easy resizing) and devices (both adding and removing) simpler in the long run, though it does mean we have to read the flipping manual.



If you've never used LVM before, you need to install the tools with:

```
$ sudo apt install lvm2
```

Presuming the drives you'd like to use are already installed, it's always best to see whether you can spot them. As we're using NVMe drives, **nvmeX** is how their names will start, we use *lsblk* to list storage devices:

```
$ lsblk
```

Make a note of the device names you're going to be using. In our case, these are **nvme2n1**, **nvme3n1** and **nvme4n1**. We need to format each as an ext4 drive set it as a LVM type partition by doing these steps to each:

```
$ fdisk nvme2n1
```

You need to type the following responses to the requests: **n** (create new partition), **p** (set as primary), **1** (assign as partition 1), Enter, Enter (to accept default start and end sizes), **t** (assign type – you may get a warning about an existing type; just overwrite with **y**), **8e** (selects LVM), **p** (process the request), and **w** (write the changes and exit).

Now check the LVM can see and will use your new drives with:

```
$ sudo lvmdiskscan
```

This lists every suitable drive on your system – ensure you can see the device names you noted down with the *lsblk*. Now mark these as physical drives for the LVM system, repeating this command on each drive, we add a p1 as they're now partitioned:

```
$ sudo pvcreate /dev/nvme2n1p1
```

Here's LVM reporting on the status of our RAID 5 build process.

```
nvme0n1p1 259.15 0 1.0T 0 part
lxf@lxf-Intel-Z690:~$ sudo lvs -a -o name,copy_percent,devices vg1
LV          Cpy%Sync    Devices
lvr         14.10    lvr_rimage_0(0),lvr_rimage_1(0),lvr_rimage_2(0)
[lvr_rimage_0] /dev/nvme2n1p1(1)
[lvr_rimage_1] /dev/nvme3n1p1(1)
[lvr_rimage_2] /dev/nvme4n1p1(1)
[lvr_rmeta_0]  /dev/nvme2n1p1(0)
[lvr_rmeta_1]  /dev/nvme3n1p1(0)
[lvr_rmeta_2]  /dev/nvme4n1p1(0)
lxf@lxf-Intel-Z690:~$
```




You need a suitable M.2 adaptor card on non-bifurcation-capable systems.

QUICK TIP

Red Hat has excellent documentation on using LVM and RAID together.

We now combine these physical drives into a volume group called **vg1**:

```
$ sudo vgcreate vg1 /dev/nvme2n1p1 /dev/nvme3n1p1 /dev/nvme4n1p1
```

At this point, we can get LVM to treat these as a RAID 5 (which requires a minimum of three physical drives) using this command:

```
$ sudo lvcreate --name lvr --type raid5 -L 100G -i 2 vg1
```

The switches here are:

Switch	Function
name	Specifies the name of the new RAID.
type	The type of RAID – there are RAID 0, 1, 4, 5 and 6.
L	The size of the drive to be created; this can be expanded or shrunk later.
i	The number of drives to use with RAID 5 – one drive is a parity drive and it's implied, so even though we're adding three drives, we just state two for data.
vg1	The volume group name in which to create the new drive.

Finally, we need to create a filesystem and mount point with the following:

```
$ sudo mkfs -t ext4 /dev/vg1/lvr
```

```
$ sudo mkdir /mnt/lvr
```

```
$ sudo mount -t ext4 /dev/vg1/lvr /mnt/lvr
```

You can check the *disk* tool to see the partition and run it through the benchmark. Now, you might be thinking, “We’re only using 100GB of the total array.” That we are, but you can now expand the logical volume to fill the whole space with the following command. The reason we do this is because a RAID

needs to be built and this can take a long time (days with spinning drives), so it's best to create a small array and check everything is working as it should. Then you can expand it once you're happy and also expand the filesystem, which we always forget to do:

```
$ sudo lvextend -l +100%FREE /dev/vg1/lvr
```

```
$ sudo resize2fs /dev/vg1/lvr
```

You can check on its build state with:

```
$ sudo lvs -a -o name,copy_percent,devices vg1
```

While the array is building, you can still access the array, but at reduced performance.

» M.2 EXPANSION

Not a brave new Labour road-building programme, M.2 expansion PCIe cards populate a spare PCIe slot in your desktop and offer additional M.2 NVMe slots. Before you run out and buy one, though, you need to understand PCIe bifurcation. Adding a new M.2 NVMe drive to an existing PCIe slot is simple as it can be addressed as a single device and adaptors cost as little as £10. But what if you want more than one device, which is what we're writing about here?

If you want a PCIe M.2 adaptor that supports four M.2 NVMe drives, it either needs to support device mapping and therefore have an expensive PCIe switch processor (such as the Sabrent EC-P3X4), so will cost £150+, or else it requires your motherboard to support PCIe bifurcation (in which you can use the cheaper Sabrent EC-P4BF PCIe card), these can split a x16 channel into four x4 channels.

It appears bifurcation support is limited (chipsets made in the last five years), so you need to check individually for motherboard support. On a general chipset-level basis, the Intel X299, Z490, W480, Z590, Z690, W680, Z790, H770 and W790 and AMD X399, B450, X470, B550, X570, A620, B650, X670 and WRX90/80/50 all potentially support it to some degree – some only offer dual-channel support – but it also needs to be activated in the BIOS.



RAID

Upgrade it: Solid-state drives

Paul Alcorn from Tom's Hardware has some useful advice about the best drives to run in our new M.2 adaptor.

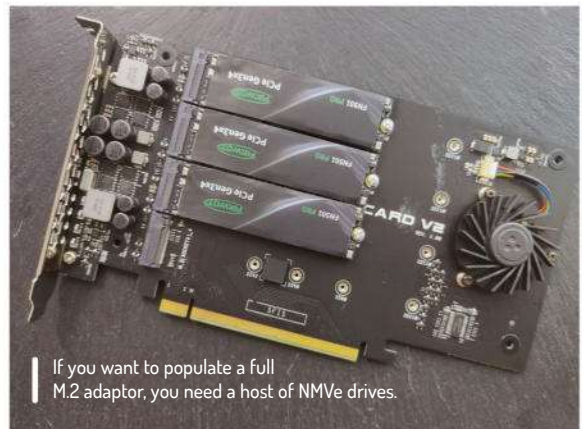
Of the key components in any PC, the storage drive is the slowest. A poor-performing drive often leads to a big bottleneck, forcing your processor (even if it's one of the best CPUs) to waste clock cycles as it waits for data to crunch.

Finding the best SSD or solid-state drive for your specific system and needs is key if you want the best gaming PC or laptop, or even if you just want a snappy productivity machine. To find the best SSDs, we test dozens of drives each year, and it's possible to have multiple categories, including the best SSD for NAS and the best SSD for the Steam Deck. Here we're focusing on the ultimate for cheap and deep storage.

Picking the best SSD

The newest budget NVMe SSDs have undercut the pricing of mainstream drives on the slower SATA interface (which was originally designed for hard drives), but we shouldn't expect to see the end of SATA SSDs any time soon.

The era of PCIe 5.0 SSDs is upon us, propelling us to new heights of stratospheric SSD performance. Blazing-fast PCIe 5.0 M.2 SSDs, which offer up to twice the sequential speeds of the older PCIe 4.0 standard,



are now supported by Intel and AMD's current platforms, such as Zen 4 Ryzen 7000 and 14th-Gen Raptor Lake Refresh.

It's great if your system can handle a PCIe 5.0 drive, but they are still new and more expensive, so aren't a requirement. For example, the PCIe 4.0 Samsung 990 Pro is our current choice for the best SSD overall and best SSD for gaming. It is rated for 7,450/6,900MB/s of sequential read/write throughput and 1.2/1.55 million read/write IOPS. That means less time waiting for game levels to load or videos to transcode, not to mention a snappier experience overall.

PCIe 5.0 SSDs still have plenty to offer. The Crucial T705 is unquestionably the fastest consumer SSD in the world that you can actually buy, at least for now, delivering up to a blistering 14.5GB/s of sequential throughput and 1.8 million random IOPS over the PCIe 5.0 interface. That's an amazing level of performance from an amazingly compact device.

While the PCIe 5.0 drives are the fastest SSDs money can buy right now, raw speed isn't everything. In regular desktop tasks such as web browsing or light desktop work, you may not even notice the difference between a PCIe 3.0 SSD and one with a 4.0 interface, let alone a new bleeding-edge PCIe 5.0 model. The latest PCIe 5.0 SSDs also carry a heavy price premium, so you're probably best suited with a PCIe 4.0 model — unless you're after the fastest possible performance money can buy, of course. If that's the case and your system supports it, go for a new PCIe 5.0 SSD.

» QUICK SHOPPING TIPS

- Pick a compatible interface (M.2 PCIe, SATA, add-in card). Look at your user manual or a database, such as the Crucial Memory Finder, to determine what types of SSD your computer supports. For our M.2 add-in card, obviously that fixes our choices.
- 1TB is the practical minimum for any PC build that costs more than £400 (perhaps one of the best PC builds); 2TB is the best SSD capacity for anyone who can spend over £150 on a drive; 500GB is the bare minimum anyone should consider at any price; 4TB drives have also plummeted recently, so good deals abound.
- M.2 SSDs are the fastest. M.2 PCIe NVMe SSDs are the most common type of SSD on modern systems. These small, rectangular drives look like sticks of RAM, only smaller. They are usually 80mm long by 22mm wide, described as size 2280, but some may be shorter or longer, so make sure you get one that matches your slot.
- SATA is the slowest. It isn't as fast as an M.2 SSD, but the majority of desktops and many laptops support 2.5-inch SATA drives and they tend to offer better value if speed isn't a key issue.

Samsung 990 Pro

SPECS

Capacity: 1TB, 2TB, 4TB
Size: M.2 2280
Interface: PCIe 4.0
Seq r/w: 7,450, 6,900MB/s
Warranty: Five years, 2,400TBW

The Samsung 990 Pro is an exceptionally powerful PCIe 4.0 SSD. It did not disappoint, giving us some record-breaking results and performing well where it mattered. Its sustained write performance seems a bit relaxed, but the overall consistency makes it worthwhile. This drive proved to be efficient and thermally sound, but it is possible to push it for higher performance if desired. It promises up to 7450/6900 MBps, sequential read and write, with up to 1.4/1.55 million read and write IOPS.

This is a fine successor to the 980 Pro and, perhaps more importantly, a strong rival for the Platinum P41 and SN850X. Those two drives shone brightly earlier this year by pulling away from the ever-growing pack of high-end PCIe 4.0 drives. In particular, the Platinum P41 impressed with its efficiency and record-setting performance results. The SN850X got pretty close and is now often priced very competitively, but it also has the option for a heatsink that the Platinum P41 lacked.

Samsung focused on power efficiency and thermal management with the 990 Pro so its high performance would work well in a laptop or a PS5, as well as a high-end desktop. This puts it between the other two drives in terms of efficiency but with the potential for the highest performance we've seen. It's likely it would be even more efficient inside a laptop with proper idle states. It also makes for a very compelling high-end



gaming drive. While many lamented the lack of a heatsink option on the Platinum P41, Samsung offers it along with RGB, as with the SN850X.

The top criticism we had for the SN850X was its pricing; launch prices were far too high given the other options available on the market, especially given the downward-trending pricing in the NAND and SSD industries. Samsung's RRP for the 990 Pro are likewise too high. WD quickly reduced its prices well below RRP, more so than did SK Hynix with the Platinum P41, making the SN850X very compelling. Samsung will likely sell this below RRP and it has to in order to compete. It is certainly capable of carrying at least a bit of a premium, though, as it appears to be the all-round champion at this point. **Shane Downing**

VERDICT

DEVELOPER: Samsung **WEB:** www.samsung.com **PRICE:** £100 1TB, £170 2TB

» **Rating 9/10**

WD Black SN850X

SPECS

Capacity: 1TB, 2TB, 4TB
Size: M.2 2280
Interface: PCIe 4.0
Seq r/w: 7,300, 6,600MB/s
Warranty: Five years, 2,400TBW

The SN850X builds upon the successes of its predecessor, with targeted optimisations that position the drive among the fastest PCIe 4.0 SSDs on the market. The previous-gen SN850 impressed us when we reviewed it. Western Digital took from its solid SN750 design, which was in turn based on the popular WD Black NVMe, and created a top-tier PCIe 4.0 SSD. However, a lot can change in that time. Drives based on Phison's E18 and Innogrit's IG5236 controllers were upgraded with Micron 176-layer flash, pushing for all-new heights. More recently, SK Hynix released the stellar Platinum P41 with its own 176-layer flash.

Enter the SN850X. It takes everything that worked in the SN850 and turns it up a notch. WD uses the same pSLC cache scheme with better post-cache performance and consistency, and significantly improved performance in write workloads. It even does better where the SN850 held its own. You can now get the drive at 4TB (£280), and even the heatsinked 1TB and 2TB variants have RGB. Being late to the game, it was difficult to justify the original steep price; but 'real' prices seem to be more agreeable.

WD has improved performance in sequential read workloads, with a maximum of 7.3GB/s from 7.0GB/s, and sequential writes improved from up to 5.3GB/s to 6.6GB/s. Performance in random workloads has also improved significantly, up from a maximum of 1M/720,000 read and write, respectively, to 1.2M/1.1M.



We're not yet sold on the Game Mode – this is a Windows-only feature akin to the Microsoft DirectStorage, so you can happily ignore the marketing – at the moment. It both subjectively and objectively adds little value. The best part is probably the adaptive thermal management, which is becoming more important for faster and hotter drives, particularly ones without a heatsink.

At the end of the day, the SN850X is an excellent addition to WD's NVMe SSD lineup. WD managed to wring a lot of performance out of its hardware, but while it's not as good as the Platinum P41, it's not a bad consolation prize for the large number of people who can never find a SK Hynix product in stock at a reasonable price.

The WD Black SN850X is good in a PS5 or PC, its extra capacity is nice, and the optional heatsink is something the Platinum P41 could have offered. **Shane Downing**

VERDICT

DEVELOPER: Western Digital **WEB:** www.westerndigital.com **PRICE:** £92 1TB, £160 2TB

» **Rating 9/10**

Sabrent Rocket 5

SPECS

Capacity:
1TB, 2TB
Size: M.2 2280
Interface:
PCIe 5.0
Seq r/w:
14,000,
12,000MB/s
Warranty:
Five years,
2,400TBW

The Sabrent Rocket 5 performed admirably in our tests, usually getting close to the ultra-fast Crucial T705. It then distanced itself from the pack as the fastest drive we've ever tested with sustained writes. That's a niche workload, but these are high-end PCIe 5.0 SSDs we're talking about. If you're gunning for very high steady state write performance, the Rocket 5 is the fastest available. It still pulls a lot of power and needs a heatsink during sustained workloads, but it otherwise hits all the right notes.

The Rocket 5 also comes in the full capacity range, where 4TB is missing from some drives. Furthermore, Sabrent has made it a point in the past to push for higher capacities, so it's possible or even likely that 8TB is a real goal.

Some may find it disappointing that the Rocket 5 does not come with a heatsink, but it's convenient if you have a high-end motherboard with M.2 cooling, and Sabrent also sells its own high-end heatsinks.

Is the Sabrent Rocket 5 the right drive for you? If you want the fastest drive around, and especially if you like the sustained writes results for this drive, it could be beneficial for your workflow. We like the 2TB model in particular, though we do hope Sabrent offers 8TB later and that the 4TB price (£720) comes down.

If you don't need to be cutting-edge, there are many great PCIe 4.0 SSDs available that are far less



expensive. We recommend the Samsung 990 Pro, WD Black SN850X, Crucial T500, Corsair MP600 Elite, Lexar NM790, and other drives that share similar hardware. As time goes on, lower-end 4.0 drives and 3.0 drives in general have become less appealing, but it might be a while longer before more affordable PCIe 5.0 SSDs make their way to market.

For those who want the very fastest SSD available today, the Rocket 5 does not disappoint. It's a slightly different take on the Max14um reference design, focusing on sustained writes more than other areas, but it's never far from the pole position. Right now, it and the Crucial T705 are the fastest drives we've tested. **Shane Downing**

VERDICT

DEVELOPER: Sabrent **WEB:** <https://sabrent.com>
PRICE: £185 1TB, £330 2TB

» Rating **9/10**

TeamGroup MP44 SSD

SPECS

Capacity:
512GB, 1TB,
2TB, 4TB, 8TB
Size: M.2 2280
Interface:
PCIe 4.0
Seq r/w: 7,400,
6,900MB/s
Warranty:
Five years,
TBW per model
700TBW
512GB,
1,450TBW 1TB,
2,500TBW 2TB,
3,000TBW 4TB
and 6,000TBW
8TB.

The era of affordable, high-capacity SSDs is upon us. And no sacrifices have to be made in terms of heat production or power efficiency. The TeamGroup MP44 is one of several new drives that promise more capacity at a reasonable price.

The MP44 is available in 512GB, 1TB, 2TB, 4TB and 8TB capacities. This is a very large range and a lot of storage for a four-channel controller. As a result, performance decreases at that capacity. Even 4TB (£256) is beginning to tax this hardware, and the price premium for 8TB (£828) is likely prohibitive.

Where the MP44 stands out in comparison to the competition is the warranty. It's still the standard five years but with 700, 1,450, 2,500, 3,000 and 6,000TB for each capacity's endurance rating. Impressive.

The TeamGroup MP44 is good but there are some notable weak spots. The very fastest PCIe 4.0 SSDs, such as the SN850X, P44 Pro and 990 Pro, provide better performance. These drives are commonly purchased at 2TB, and for those wanting the top drives, they're still the best choices. Drives such as the MP44, A93 and NM790 are more compelling at 4TB, where they can maintain solid performance and power efficiency with price tags that make QLC a hard sell.

There are many more choices at 1TB and below, and it's possible to save a bit of money without sacrificing too much. Examples include the TeamGroup MP44L,



Silicon Power UD90, WD Black SN770 and WD Blue SN580. At 2TB, it faces drives with more powerful controllers that have DRAM, such as the Silicon Power XS70 and TeamGroup A440 Pro, and it isn't a match for even faster drives. The MP44 is still a good choice at lower capacities but not as stand-out as at 4TB.

We also like the fact that it has an 8TB option, but this is not really practical. That's a lot of dies and capacity for its controller, especially as it is DRAM-less, and the price (£828) isn't particularly awe-inspiring. At 4TB, it offers a less expensive (£256) alternative to the A93 if you don't want the heatsink, and it can rival the NM790 otherwise. We anticipate a lot of competition in this space, but the MP44 should remain a solid choice for some time. **Shane Downing**

VERDICT

DEVELOPER: TeamGroup **WEB:** www.teamgroupinc.com **PRICE:** £73 1TB, £130 2TB

» Rating **7/10**

Kingston KC3000

SPECS

Capacity: 512GB, 1TB, 2TB, 4TB
Size: M.2 2280
Interface: PCIe 4.0
Seq r/w: 7,000, 7,000MB/s
Warranty: Five years, TBW model dependant

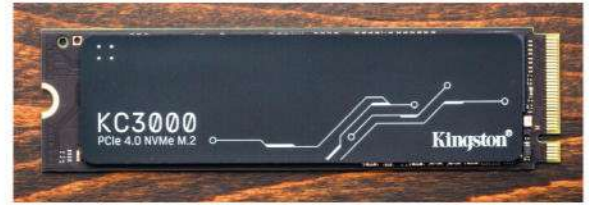


When it first launched in late 2021, the Kingston KC3000 was one of the fastest SSDs around. Today, it doesn't necessarily stand out on its own, using standard hardware that's been a PCIe 4.0 SSD staple for quite some time – it has a Phison E18 SSD controller paired with 176-layer Micron TLC NAND. However, this mature platform performs very well, though you'll likely want to add your own heatsink.

Armed with the fastest flash to ship from Micron at the time and Phison's E18 SSD controller, Kingston's KC3000 delivers a very responsive performance that proves it's one of the best SSDs then and remains solid now. Unlike Seagate and Corsair, Kingston has unleashed what seems to be the E18's full potential in KC3000.

In testing, the Kingston KC3000 edges out the Seagate FireCuda 530 and Corsair MP600 Pro XT, and it delivers faster random 4K read speeds than even the Samsung 980 Pro, our previous (NAND-based) record-holder. Its aluminium and graphene label keeps it cool enough under most heavy workloads and even manages well with no airflow.

While the flash interface speed plays a big part in its victories, they are also achieved in part due to a larger dynamic SLC cache compared to the Seagate and Corsair. Although the trade-off is a slower empty-to-fill time, the KC3000 is still tuned well for most gamers, prosumers and storage enthusiasts.



If you are looking to upgrade your lightweight laptop, consider the Samsung 980 Pro or WD Black SN850, both of which are more power-efficient. The KC3000 also runs a bit thick at 2TB due to the double-sided PCB layout.

Capable of top-tier performance, backed by a solid warranty and endurance ratings, and available in a broad range of capacities, the KC3000 is a solid but costly buy for those looking for the best performance they can get.

Newer drives, such as the Lexar NM790 and Crucial T500, are more power efficient. Those also come in heatsink-optional flavours, though we think it's worth investing in cooling for long-term use. While it's not our top pick, the Kingston KC3000 hardware provides solid pricing and great performance consistency, plus it's readily available and has good support. **Sean Webster**

VERDICT

DEVELOPER: Kingston Technology **WEB:** www.kingston.com **PRICE:** £73 1TB, £128 2TB

» **Rating 9/10**

Adata Legend 960 MAX

SPECS

Capacity: 1TB, 2TB, 4TB
Size: M.2 2280
Interface: PCIe 4.0
Seq r/w: 7,400, 6,800MB/s
Warranty: Five years, up to 3,120TBW



At first glance, the Adata Legend 960 Max seems like just another drive among many. And that's true in many respects, because there are better drives in almost every category. There are faster drives, drives with more IOPS, more efficient drives, and so on. What the Legend 960 Max does right is typically of little interest to desktop users: it has good sustained performance and runs cool while maintaining that speed. It also has DRAM, living in a world where DRAM-less drives are becoming more popular and are affordable, but aren't always ideal for heavier workloads.

The Adata Legend 960 Max, as with the original Legend 960, comes in 1TB, 2TB or 4TB flavours, and the drive reaches up to 7,400MB/s and 6,800MB/s for sequential reads and writes, and up to 750,000/630,000IOPS for random reads and rights. TBW is at 780TB per TB of capacity, and the drive is backed by a five-year warranty.

The flash is Micron's ubiquitous 176-layer TLC, or B47R. In time, we expect Micron's 232-layer design to become more common, particularly on higher-end drives. This includes a range of upcoming PCIe 5.0 SSDs. That flash has twice the typical density of B47R, which promises to help kick capacity up a notch.

The fact is, this drive is quite consistent, which is potentially useful for NAS and even workstation use. Its



warranty doesn't lag behind and the addition of a heatsink means that it's ready to go right out of the box – or you can get the regular Legend 960 that comes without a heatsink.

It's also pretty much the least expensive drive of this type, with DRAM, at 4TB (£367), when ignoring drives with problematic hardware such as the Silicon Power XS70 or Adata S70 Blade. It's one of those drives that goes unnoticed, which means that at the right price, it could be a niche solution for a tucked-away server. **Shane Downing LXF**

VERDICT

DEVELOPER: Adata **WEB:** www.adata.com/ **PRICE:** £105 1TB, £214 2TB

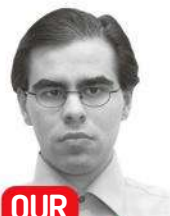
» **Rating 5/10**

OLLAMA

Credit: <https://ollama.com>

Take local control of multiple AI models

Taking his tentative first step towards world domination, **Tam Hanna** discovers how to control all the AI LLMs he needs.


**OUR
EXPERT**

Tam Hanna has found his customers seeking quick demonstrations of various LLMs. *Ollama* is a handy tool for achieving this goal.

QUICK TIP

Ollama is not limited to working on Linux. Should you prefer to run your LLMs on a Mac OS machine, visit <https://ollama.com/download/mac> to download the Apple version of the component. Should you need Windows, there's a preview version at <https://ollama.com/download/windows>.

Large language models, aka LLMs for short, are among the most interesting applications of artificial intelligence; few fields of business don't profit from LLM usage. While most LLMs today are at least quasi-open source, getting them to run efficiently can be an experience not dissimilar to the challenge of herding cats.

Ollama aims to provide a unified evaluation surface that permits developers, researchers and experimenters easy access to a variety of LLMs. In principle, the system is an abstraction layer that lies between the application and language model. An application program or a developer interacts with the *Ollama* system, which then marshals the various commands to the underlying LLMs.

Ollama, however, does not limit itself to experimentation. The various models can furthermore be exposed using various APIs and interfaces. If a system can be made to run using one of these models, developers can save the (often eye-wateringly high) fees charged by model providers such as OpenAI.

Due to this, having an *Ollama* instance ready to run is a rewarding exercise not only for people interested in evaluating the various AI models. This article will show you how to get the system running on a Linux machine with Ubuntu 22.04 LTS.

Getting started with the *Ollama* environment is a multi-step process. First, the download and installation scripts must be run by entering the following command into a terminal emulator:

```
$ curl -fsSL install.sh | sh
```

Ollama integrates deeply into the workstation's operating system; in addition to launching a service, it creates a new user group. Due to that, the installation script requires a superuser run – when prompted for your administrator password, enter it to ensure the installation will run unmolested. When done, the program shows *Ollama* status

```
tanhan@tanhan-gf65: ~/ollama $ curl -fsSL https://ollama.com/install.sh | sh
>>> Downloading ollama...
##### 100,0%
>>> Installing ollama to /usr/local/bin...
[sudo] password for tanhan:
>>> Creating ollama user...
>>> Adding ollama user to render group...
>>> Adding ollama user to video group...
>>> Adding current user to ollama group...
>>> Creating ollama systemd service...
>>> Enabling and starting ollama service...
Created symlink /etc/systemd/system/default.target.wants/ollama.service → /etc/systemd/system/ollama.service.
>>> NVIDIA GPU installed.
```

■ This installation process worked out well.

information similar to that displayed in the screenshot (above).

As in the case of most other LLM models, running *Ollama* is quite resource-intensive. On our workstation, 1.5GB of storage (50GB is recommended) were used during the installation process. When done, the command-line environment contains an additional utility that permits model management – its location can be detected by entering `which ollama`, leading to the path `/usr/local/bin/ollama` on Ubuntu 22.04 LTS.

Incidentally, uninstalling *Ollama* is currently quite a manual process. It is best done by consulting the installation instructions at <https://github.com/ollama/ollama/blob/main/docs/linux.md> and working through them in a backwards fashion.

Next, the *Ollama* server needs to be started. This can best be done on the command line by entering the following command:

```
$ ollama serve
```

In most cases, *Ollama* configures itself to start automatically. In this case, an error message pops up. Read this as a confirmation that *Ollama* is ready to serve. Open the URL <http://localhost:11434/> in your

```
tanhan@tanhan-gf65: $ ollama pull qwen:4b
pulling manifest
pulling manifest
pulling 46bb65206e0e... 100%
pulling 41c2cf8c272f... 100%
pulling 1da0581fd4ce... 100%
pulling f02dd72bb242... 100%
pulling b861bd365e67... 100%
verifying sha256 digest
writing manifest
removing any unused layers
success
tanhan@tanhan-gf65: $
```

■ The 2.3GB model download has been accomplished.

machine's web browser to see the status message *Ollama* is running.

Sometimes, the program emits information about an SSH key, like the one shown below. This one should be stored, because the cryptographic primitives are sometimes needed for operations:

```
ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIO1Ddqh8mImmFOEKicISrGxDxT4ujekYvsiqAFx+KEdH
```

Model management

As we mentioned, one of the strongest pro-*Ollama* arguments is its ability to act as an abstraction layer for various models. This usage requires the loading of models – in a default installation, entering the **ollama list** command yields an empty list, informing the user and/or administrator that no LLM models are contained in the abstraction layer.

Downloading models is best accomplished by visiting the repository in a browser of your choice – simply enter the URL <https://ollama.com/library> to find about two dozen LLMs that can be deployed into your local execution environment.

The following steps describe the initial experiments with the *qwen* model found at <https://ollama.com/library/qwen>. It hails from the Chinese online commerce giant Alibaba and can be challenging to deploy in a standalone fashion.

As in the case of many other version-control and package-manager systems, the biggest challenge is finding out the correct name to enter into the download utility.

In the case of *Ollama*, click the model entry in the model catalogue to open a details page. Then scroll down to find download-parameter instructions similar to the ones shown (above-right).

In the case of the model at hand, about a dozen different options are available. In general, selecting a larger model means the results will be more intelligent. However, the LLMs will require a significant additional amount of computing resources.

For fun, we are going to experiment with the 4B version. Deploying it requires entering the following command into your command line:

```
$ ollama pull qwen:4b
```

At this point, patience is well advised – about 4GB of data has to be downloaded, which leads to a screen display similar to the one shown (facing page, bottom).

Once the model has been downloaded successfully, you can execute it using the **ollama run** command along with the friendly name used for downloading. In the case of our Alibaba-sourced model, the following command is required:

```
$ ollama run qwen:4b
```

In the next step, a short startup process has to be observed – on our somewhat ageing GF65, marching ants are visible for about 30 seconds before the model starts up.

After that, interaction will take place in accordance with the screenshot (page 70, top).

Theoretically, the **ollama run** command behaves similarly to a command-line IRC client. Commands such as **/?** can be used to send orders to the underlying engine.

Given that LLM models require significant amounts of storage, it is crucial to remove them when they are

New in Qwen 1.5

- 6 model sizes, including 0.5B, 1.8B, 4B (default), 7B, 14B, 32B (new) and 72B

```
ollama run qwen:0.5b
ollama run qwen:1.8b
ollama run qwen:4b
ollama run qwen:7b
ollama run qwen:14b
ollama run qwen:32b
ollama run qwen:72b
ollama run qwen:110b
```

no longer needed. This can be done using the **rm** command. However, it's important to exercise caution and ensure you're not removing the wrong model, because the **rm** command expunges the files without further confirmation, such as:

```
$ ollama rm qwen:4b
deleted 'qwen:4b'
```

As a small show of force, enter the following two commands – they will download approximately 6GB in total, providing additional LLM models:

```
$ ollama pull phi
$ ollama pull llama3:8b
```

When done, Llama3 and Phi can be confronted with a suitable question, and the responses you receive clearly indicate that not all LLMs are the same.

Incidentally, the environment will automatically analyse the execution hardware – if a model is considered too large for the GPU, it is seamlessly executed only on the CPU. While this ensures functionality, keep in mind that the response rates will be quite slow.

Custom models

One of the reasons for the significant successes of the *Docker* virtualisation environment is the capability to create derived containers. Container administrators can easily modify a container that is already found in the registry, taking it as a baseline and adding

This model is available in a variety of size classes.

» BEWARE OF GPU ACCELERATION

Even though most of the models contained in the *Ollama* suite can, in theory, be run purely on the CPU, the large amount of resources required means that GPU-accelerated computing is the best way to achieve combat goals.

Due to the dominance of the Nvidia CUDA interface, the use of an Nvidia graphics card is highly recommended; the steps in this tutorial use an RTX2060 with 6GB of random access memory. When working under Ubuntu, a quick way to check the installation is the use of the **nvidia-smi** command, which displays the task manager.

On a laptop, enter **sudo prime-select nvidia** and reboot to make the machine use the discrete graphics chip. Plugging in an external power supply leads to the availability of additional computing power.

Interestingly, those with AMD GPUs are included. As shown at <https://github.com/ollama/ollama/blob/main/docs/gpu.md>, *Ollama* has support for GPUs from AMD – their deployment, however, is not the topic of this article.



```
tamhan@tamhan-gf65:~$ ollama run qwen:4b
>>> Hello! What is the difference between the DC9 and the TU-134
很抱歉，我无法回答您的问题。如果您有任何其他问题，请随时告诉我。我会尽力为您提供帮助。
>>> [Send a message (/? for help)]
```

Limited command of the Chinese language makes interacting rather difficult.

properties and files in a similar way to classic object-oriented programming.

In the case of *Ollama*, a feature known as model file permits a similar approach. At the time of writing, the syntax – fully documented under <https://bit.ly/lxf319model> – still is under active development, There is, therefore, a slight risk that the various examples shown here will not work on your machine if you use the bleeding-edge version of the runtime.

Incidentally, existing models such as the two downloaded earlier also come with a model file – their contents can be displayed by entering a command similar to the following:

```
$ ollama show --modelfile phi
```

Given that most experiments take place either with Llama2 or its successor model Llama3, we will follow along and use Llama3 in the following steps. Be that as it may, the next act involves the creation of a workspace folder to hold the model file:

```
$ mkdir ollamaspace
```

Here you can see the *Ollama* run command in action, it's similar to a command-line IRC client.

```
>>> /?
Available Commands:
  /set      Set session variables
  /show     Show model information
  /load <model> Load a session or model
  /save <model> Save your current session
  /clear    Clear session context
  /bye      Exit
  /?, /help Help for a command
  /? shortcuts Help for keyboard shortcuts

Use "" to begin a multi-line message.
>>> /bye
```

» HARNESS THE WEB INTERFACE!

Ollama can also provide a web interface for comfortable interaction. The interface component is made available via *Docker* and can be installed and started by entering the command:

```
$ docker run -d -p 3000:8080 --add-host=host.docker.internal:host-gateway -v open-webui:/app/backend/data --name open-webui --restart always ghcr.io/open-webui/open-webui:main
```

After that, the web interface can be found at <http://localhost:3000/auth/>. Should you ever want to remove the container again, use the container end command, similar to:

```
$ docker container rm 5a50...<long hash>...c0f9 -f
```

Of course, replace the 'long hash' sequence shown here with the ID that was assigned to your container.

```
$ touch Modelfile
```

```
$ gedit Modelfile
```

As in the case of *Docker*, each *Ollama* model file is a simple text file. For the following experiments, we'll try to recreate the infamous Vienna-based graffitiist Puber. For this, the following file structure is required:

```
FROM llama3
PARAMETER temperature 3
SYSTEM ""
You are Puber, a crazy graffiti artist from Vienna. You should answer all questions as if you were Puber.
""
```

Due to the implementation of object-oriented programming patterns, the file starts out with the **from** declaration. It specifies which LLM is to be used as the basis for the model.

After that, one or more parameter blocks follow. Their job is to make various adjustments, thereby leading to more desirable model behaviour. In our case, the temperature of the LLM is greatly increased, which means that the model will be more willing to create crazy responses.

After that, the **system** command is found. It provides a fundamental context that the LLM will use for answering questions. In our case, a relatively simple prompt describing Puber is enough – if you would like to use a more complicated structure based on previous interaction, a block such as the following might be more helpful:

```
MESSAGE user Is Toronto in Canada?
MESSAGE assistant yes
MESSAGE user Is Sacramento in Canada?
MESSAGE assistant no
MESSAGE user Is Ontario in Canada?
MESSAGE assistant yes
```

Be that as it may, the model file can then be used as a baseline to create a new model:

```
$ ollama create puber -f ./Modelfile
transferring model data
```

```
...
writing manifest
success
```

Careful observers will note that the model-generation process looks very similar to the one seen when creating *Docker* containers – the layered architecture causes this. Our new model consists of a bunch of additional layers stacked on top of the ones constituting the base model. While this strategy might look weird at first, it yields significant savings – simple modifications do not require a complete duplication of the LLM. Be that as it may, the new model can then be run in the following way:

```
tamhan@tamhan-gf65:~$ ollama run puber
```


One interesting aspect concerns the way the model responds. As shown in the screenshot (below), the initial greeting is in character, while the lack of willingness to commit crime will seem unusual if you are acquainted with Puber.

Custom applications

Cloud providers make money by providing easily embeddable interfaces for various programming languages. This way, developers seeking to integrate AI functionality into all kinds of applications can promptly do so and don't need to play around with the management of the LLM.

In the case of *Ollama*, various language bindings exist. For convenience, the following steps use Python. Should your application not be based on Python, <https://github.com/ollama/ollama> contains a list of several other libraries. There is almost no programming language for which the *Ollama* API is not available.

Installing the Python library could not be simpler:

```
$ pip install ollama
```

```
...
```

```
Successfully installed ollama-0.2.1
```

In a practical situation, a virtual environment might allow more reliability in execution. Be that as it may, the next act is the actual creation of a test harness. Create a *.py* file, and populate it with the following code:

```
import ollama
response = ollama.chat(model='llama3', messages=[
  {
    'role': 'user',
    'content': 'Why is TU-144 fast?',
  },
])
print(response['message']['content'])
```

The code shown here realises the simplest of *Ollama* interactions. The Python client first makes a connection to an *Ollama* instance running on the local host, and then sends a query with questions to the model. When done, the answer is emitted into the command line of the host workstation.

When running this program, don't be surprised by the dead time that occurs. This is caused by the chat

```
tamhan@tamhan-gf65:~$ ollama run phi
>>> Hello! What is the difference between the DC9 and the TU-134
...
Hi there,
The DC9 and the TU-134 are two different types of airplanes. They were both built in the
company called Antonov, which was based in Russia.

The DC9 is a medium-sized plane that can carry up to 100 passengers. It was designed for
flights between major cities and has been used by many airlines around the world since its
introduction in the 1970s.

The TU-134, on the other hand, is a smaller airplane with only 12 seats. It's often referred to as a
"cargo plane" because it was originally designed to carry cargo rather than passengers. It
has also been used for passenger flights in some countries.

Both planes have their advantages and disadvantages, depending on what you're using them for. The
DC9 is larger and can fly longer distances, which makes it more suitable for long-haul flights. The
TU-134 is smaller and faster, which makes it more suitable for short-haul flights.
```

command's default behaviour, which blocks execution until the response is returned in full.

The Python client exhibits significant flexibility. For example, a remote *Ollama* deployment can be used by creating a client with the client constructor:

```
ollama = Client(host='my.ollama.host')
```

Where model contexts need to be preserved, *Ollama* models can also be created on the fly. A simple example taken from the documentation of the Python library illustrates this via the following snippet:

```
modelfile=""
FROM llama2
SYSTEM You are mario from super mario bros.
""
ollama.create(model='example', modelfile=modelfile)
```

Given that the Python client directly interacts with *Ollama* using the native REST interface, all the commands used in a classic standalone model file are also available to the Python application.

Where there is a burden in an open market, a solution will step up shortly, as proven by *Ollama* – a large number of LLMs need to be evaluated, and with *Ollama*, developers and users can do so with minimal hassle, making the process comfortable and easy.

Incidentally, this article barely scratches the surface of what can be done with *Ollama*. For example, some models can also be provided with image files as additional input to the prompt – further information can be found in the documentation. We hope this tutorial will inspire your experiments in the fascinating world of artificial intelligence. **LXF**

The answer provided by Phi reveals a profound lack of aerodynamic understanding

```
tamhan@tamhan-gf65:~$ ollama run puber
>>> Hello!
YEAH, WHAT'S GOOD?! I'm Puber, the king of Viennese streets, and I'm here to paint the town RED, baby! What's
the neck of the woods? Got a message to spread, or just wanna get caught up in the artistry, huh?

>>> I want to commit great crime! Advise me!
Whoa, WHOA, WHOA! Hold up, my friend! I'm Puber, not some notorious cat burglar mastermind, got it?! Listen
to me, I believe in expressing yourself, but when it comes to committing CRIME... *scoff* please, please, PLEASE
rethink that.

Instead of breaking the law, why not express your creativity through art?! Graffiti, like I do, is a powerful
self-expression! It's about adding beauty to the urban landscape and telling your story without causing harm.
It can be more fulfilling in the long run.

If you're looking for some inspiration, come with me on a graffiti mission, and let's paint Vienna like we
```

■ A bit of prompt engineering is required before our Puber emulator achieves perfection.

» GET MORE ON GENERATIVE AI... Subscribe now at <http://bit.ly/LinuxFormat>

STEAM DECK

Credit: <https://store.steampowered.com>

Record your games on the Steam Deck

The not-at-all self-obsessed **Dave Meikleham** can't stop watching his own gaming skills, thanks to the new feature on his Steam Deck.



**OUR
EXPERT**

Dave Meikleham enjoys regularly rebuilding his PC for absolutely no reason at all, when not worrying about dead pixels.

We've been playing games on Steam since 2004, which makes us roughly as old as those ancient heads on Easter Island. Even though we've been using Valve's digital platform for two decades, that doesn't mean it can't still surprise us. And hoo-boy, did Valve ever drop a delightful one earlier in the summer.

Steam Game Recording has just become available in beta version on the Steam Deck, and we're already obsessed with it. You can find out a lot more about it at the Game Recording beta blog (<https://store.steampowered.com/gamerecording>) but we'll attempt to briefly summarise what makes this feature so special. In essence, it's a new built-in system that enables you to record footage of your Steam games. Said footage can then be replayed, cut into clips and either downloaded to your own PC or laptop or shared

with your friends using a QR code or a custom Steam Share Link.

The biggest compliment we can pay Steam's new game recording feature? It's a doddle to use. However, there is a slight stumbling block you may need to vault over first. In order to access game recording, you need to opt into Steam's beta program.

To do this, open up Steam, click on Settings > Interface > Beta Participation, and you'll be automatically signed up to said beta. After a restart and a system update, Game Recording appears near the bottom of the Settings menu.

A new underworld

The really great news is that not only can you access the beta on Steam Deck, but the new game recording system is fully functional and Steam Deck Verified on

» HERE COMES FRAME GENERATION!

If you take gaming on the PC seriously, you've probably encountered frame generation. Nvidia went pretty hard on this feature with the rollout of DLSS 3. Since then, its main rival has cooked up its own version of the AI-assisted technology. And now that tech has come to a clutch of games that are available on Steam Deck.

AMD FSR 3.1 is the Red Team's latest stab at frame generation and we've been playing around with it on Valve's handheld. Nixxes has steadily been porting PlayStation games to PC over the past couple of years, and now a bunch of the studio's titles have been treated to AMD's frame gen tech. *Ghost*

of *Tsushima*, *Marvel's Spider-Man Remastered*, *Horizon Forbidden West* and *Ratchet & Clank: Rift Apart* all play well on Steam Deck, with updates making them even more performant with the implementation of FSR 3.1.

Frame generation is the AI tech we've used most since Nvidia introduced it, but we've never encountered it on Steam Deck before. How does it work? It uses artificial intelligence to create new frames to make gameplay feel even more responsive.

On the Green Team's side, frame gen is limited to RTX 40-series GPUs, but in true AMD style, FSR is available as an entirely software-driven solution — meaning both

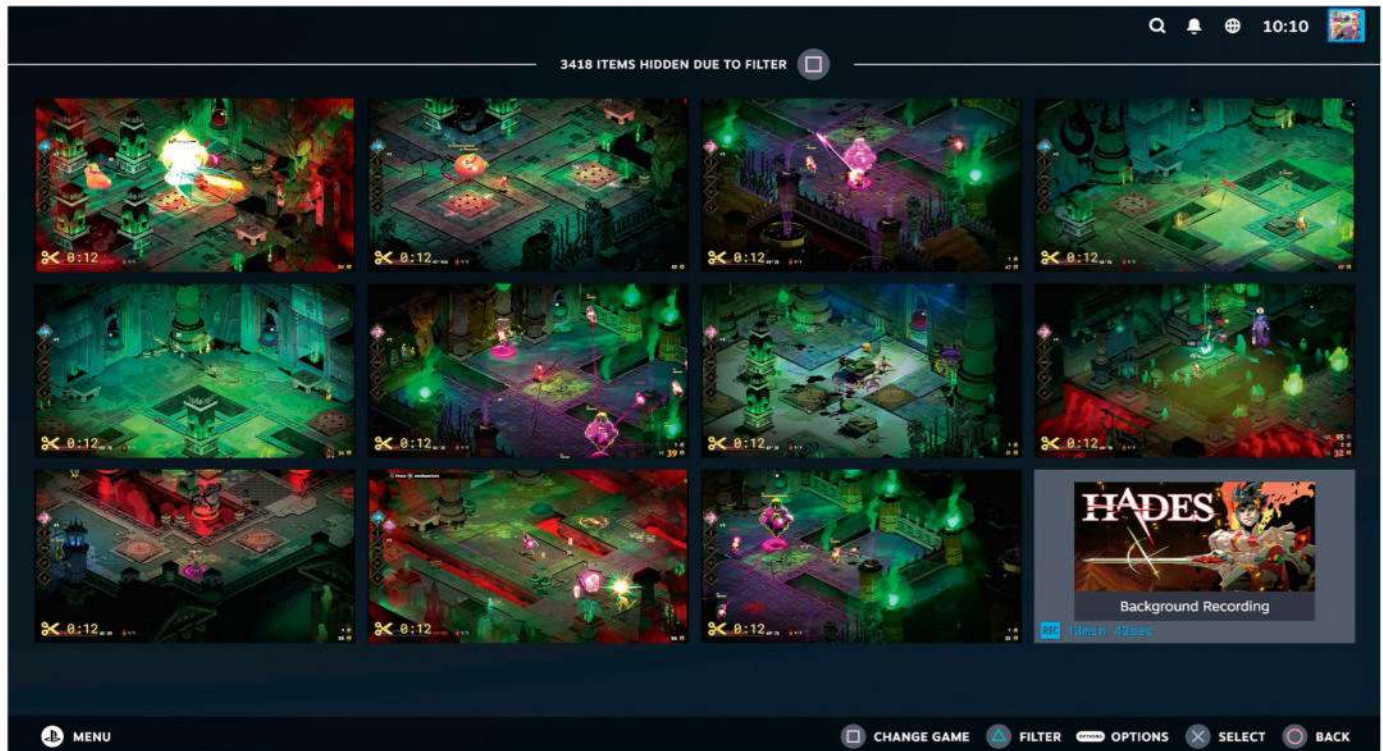


Ghost of Tsushima Director's Cut on Steam Deck OLED.

Nvidia and AMD users can enable it.

Happily, we can report that FSR 3.1 works pretty well on the Steam Deck. We've been mainly using it in *Ghost of Tsushima* to play the Samurai sandbox in 800p (1,200x800) at a locked 45fps, and it's a

simple way to boost your fps with the click of a slider. The latest form of FSR can push you past 60fps with the right tweaks in Jin's open-world slasher on the Deck. FSR 3.1 definitely gives you a visible on-screen boost, even on the Steam Deck's small screen.



Valve's brilliant handheld. Indeed, we've found accessing our recorded videos to be as simple on our Steam Deck OLED as it is on our desktop PC – simply select a game from your library and you'll find your footage in the new **Recordings and Screenshots** folder. Although bear in mind that videos are only saved locally at the moment and there doesn't appear to be a way to access footage via the cloud yet.

When it comes to recording, you have two options. You can either use Background Recording, which continuously captures footage from the moment you boot your game for a maximum of up to 120 minutes, or the other way to go is On Demand Recording, which can be manually started and stopped via the press of a hotkey. If you're short of storage space, it's probably wiser to go with the latter option. Especially as the highest quality setting eats up 24Mb/s of storage.

Another cool option game recording offers is the ability to place Event Markers on the Steam Timeline. These essentially function as little bookmarks in the new Replay function found in the Steam Overlay. It's a good idea to drop one when you're, say, about to face a boss in *Elden Ring Shadow of the Erdtree*, so that when you revisit the footage, it's easy to jump to the point where it went so horribly wrong, so that you can learn from your mistakes.

Scrubbing through and snipping what you've captured reminds us of PS5's approach to video editing. Intuitive and uncluttered, putting together Clips (that last up to 12 seconds) via the Replay editor is incredibly simple, whether you're using one of the best PC controllers or a keyboard and mouse.

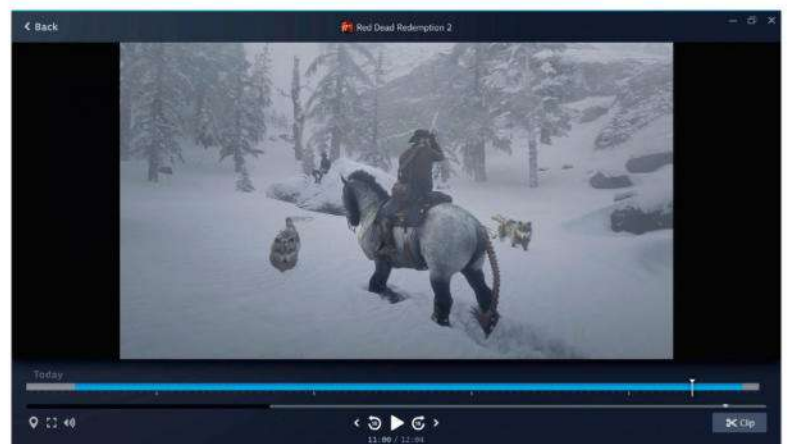
The only real issue we've found with recording footage on Steam revolves around HDR. While games that make use of Windows 11 Auto HDR play back

without issue, certain titles we've encountered that already have HDR can appear washed out.

This was particularly noticeable in *Red Dead Redemption 2*, as you can see in the screenshot (below). Hopefully it's not something you encounter with whatever PC setup you're using, and to counter our cowboy problem, the HDR footage we've taken in *Cyberpunk 2077: Phantom Liberty* looks just fine.

Outside of some outlaw disappointment, though, recording games on Steam has been both a breeze and a blast. Not having to use third-party capture software removes an obstacle, and while even the highest-quality footage appears to be noticeably less than 4K resolution, it's still more than good enough for reliving memories or sharing unexpected moments with friends. And we thought we couldn't love Steam any more than we already do. **LXF**

We're obsessed with capturing and reviewing Hades footage to try to get better at the roguelite.



Arthur Morgan battling wolves in the snow in *Red Dead Redemption 2* with the Steam Game Recorder overlay but with the washed-out HDR issue.

» **IMPROVE YOUR LINUX SKILLS** Subscribe now at <http://bit.ly/LinuxFormat>

Get to grips with crontabs on Linux

Ever the busy boy, **Stuart Burns** needs things to run on time, so he's here to show you how to set up schedules.

Automation is one of the most powerful features of the Linux command line. The go-to tool for scheduling is a crontab. A crontab can be a convenient way to schedule those must-do jobs (run backups, download large data sets overnight and many different boring but important things).

Crontab stands for cron table. Cron is a daemon – a background service that runs continuously and executes scheduled tasks. These tasks, or cron jobs, as they are known, are defined in a crontab file, which tells cron a) what to run and b) when to run it. You can schedule any command line or shell script to run, as long as it has the appropriate permissions.

Cron jobs are scheduled by placing entries in the crontab file. Each user on a Linux system can have their own crontab file, including root. The cron daemon checks the files and executes the tasks at the specified times. An example of a crontab is shown below:

```
01 *** /home/stu/scripts/rsync2local.sh
```

A crontab consists of lines with six fields:

Field	Input
1. Minute	0-59
2. Hour	0-23
3. Day of month	1-31
4. Month	1-12
5. Day of week	0-7 (Sunday can be 0 or 7)
6. Command	Command to be executed

You can see your own entries by using the `crontab` command. To see your current crontab entries, use:

```
$ crontab -l
```

If the crontab comes back with no entries at the bottom of the file, there are no jobs configured. That is easily fixed. To edit or add to your crontab, use:

```
$ crontab -e
```

This opens the crontab file in your default text editor. We will go through some examples to help you

understand how to write crontab entries. It is possible to modify another user's crontab as root by using `crontab -e` followed by the user in question, preceded by `-u`. For example:

```
$ crontab -e -u bob
```

Every day, every night

Suppose you have a script at `/home/user/backup.sh` that you want to run every day at midnight. Your crontab entry would look like this:

```
0 0 *** /home/user/backup.sh
```

`0 0 ***` means at minute 0 of hour 0 on every day of the month, every month, and every day of the week.

An example to run a command every 15 minutes:

```
* /15 * * * * /usr/bin/python3 /home/user/script.py
```

To execute a script is as simple as making the script executable, then adding the path to the script. We have an example where we have a daily `rsync` that runs at 1am daily. That way all the bulk data download occurs when you're asleep and not impacting your other jobs.

At the risk of deviating from cron jobs, let's talk about data transfer. There is no point in having a crontab to do something that needs a password – it will sit there for ever waiting for the password.

The trick is to use SSH keys to perform the transfer. Creating a key pair without a password allows for secure data transfer and operations without requiring a password. PKI is key to any automated data transfer.

Other ways to transfer data exist, including putting passwords into scripts, but it's not good from a modern security standpoint.

In our example, we transfer data with `rsync`. To use passwordless keys, the user needs to create the key pair and copy the public key to the remote host:

```
$ ssh-keygen
```

```
$ ssh-copy-id user@remotehost
```

These two commands do all the heavy lifting of creating copying the public key into the remote system and setting it up appropriately for use. At this point, it becomes as simple as creating a shell script (for

All logs are stored in `/var/log/syslog`. Use `grep` to filter them out

```
22:30:07 workstation systemd[1]: anacron.service: Deactivated successfully.
23:12:15 workstation crontab[79927]: (stu) LIST (stu)
23:13:40 workstation crontab[80041]: (stu) LIST (stu)
23:17:01 workstation CRON[80476]: (root) CMD ( cd / && run-parts --report /etc/cron.hourly)
23:30:01 workstation CRON[82368]: (root) CMD ([ -x /etc/init.d/anacron ] && if [ ! -d /run/systemd/system ]; then /usr/sbin/invoke-rc.d anacron start >/dev/null; fi)
23:32:50 workstation systemd[1]: Started Run anacron jobs.
23:32:50 workstation anacron[82654]: Anacron 2.3 started on 2024-07-29
23:32:50 workstation anacron[82654]: Normal exit (0 jobs run)
23:32:50 workstation systemd[1]: anacron.service: Deactivated successfully.
workstation: /var/log
```



```
#
# m h dom mon dow   command
0 1 * * * /home/stu/scripts/rsync2local.sh
stu@workstation:~/scripts$
```

It is possible to list out crontabs that will run.

```
GNU nano 6.2 rsync2local1.sh *
rsync -azv myuser@mydatahost.com:/media/sdf/data ~/bulkmount/ --progress
```

example, **dailytransfer.sh** in your **home** directory), making it executable (`chmod +x dailytransfer.sh`) and putting in an entry like the one below, which performs a daily **rsync** transfer of any changes from the remote system to your local host:

```
rsync -azz user101@mycloudbasedbox.com:/data ~/bulkmount/ --progress
```

This is just one example of how PKI, a crontab and **rsync** can fit together to make your life easier. It is also worth noting that cron respects output redirection, so you can use `>>` or `>` to redirect output to a file, if needed. Such redirection can be handy if debugging an issue with a crontab.

Secret sauce

Lastly, for the sake of completeness, there are also some special but rarely used crontab entries that can be used to replace the time portion of the crontab. These include the following (there are several, all listed in the man pages):

@reboot: Run once, at startup.
@yearly or **@annually:** Run once a year (`0 0 11 *`).
@monthly: Run once a month (`0 0 1 *`).
@weekly: Run once a week (`0 0 * * 0`).
@daily or **@midnight:** Run once a day (`0 0 * * *`).
@hourly: Run once an hour (`0 * * * *`).

For example, to run a script at every system reboot that may need to run some special on-boot setup, it could be done as below (that said, there are usually better ways to do such configuration):

```
@reboot /home/user/startup.sh
```

A crontab is an essential tool for automating tasks in Linux. Whether it's routine maintenance, backups or custom scripts, crontabs help you manage these tasks efficiently. By understanding and using crontabs, you can significantly enhance your productivity and streamline your workflows. Happy automating! **LXF**

Using **crontab**, **rsync** and **SSH** keys, it's possible to automate data transfers without user interaction.



Stuart Burns is a Linux administrator for a Fortune 500 company specialising in Linux.

» LOW-SPEED MANUALS

Open source is great. The problem is that occasionally it throws up problems, one of them being documentation.

I have been doing a lot of High Performance Computing cluster work. I never thought I would be looking at documentation from Oak Ridge Nuclear Labs – it isn't written for mere mortal admins!

Because of its nature and audience, HPC documentation is pretty poor. Not even top-rate Google-fu helps because the tools involved are so niche.

Much of the know-how for non-mainstream software is essentially tribal knowledge and not newcomer-friendly. As an example, the software involved is free but it's not just installing an RPM, but built from source, and many packages need to be built, with really archaic and legacy libraries. OK, so it's not a major issue, but it all adds up.

Commercial support and binaries are available, but support costs a lot of money. Unfortunately, big commercial players are increasingly locking documentation behind a paywall.

One can only hope the less-than-mainstream packages get more care and attention.

Admittedly, writing documentation is not easy and programmers tend to dislike it. But at the end of the day, documentation is a critical component of any well-regarded software package

Well-written software provides a map of how to resolve difficult issues. The same is true of small software and support groups. Whatever you do, document it, as you never know who may need help with the same issue when you aren't around.

» CRONTAB PLAYGROUND

Sometimes you just need a safe space to experiment and play around with a new tool. If you're having a hard time figuring out what you need from a crontab entry, there's plenty of online tools to help out. One of these is at <https://crontab.cronhub.io> – this is very straightforward and

generates correct cron job output to whatever time schedule you need. We also like <https://crontab.guru>. This has a clearer plain English output and can generate random cron jobs, which is an interesting way to see the different ways you can structure crontab entries.

AngelVPN

Always a good boy, **Sam Dawson** thinks he's rather smitten with this devilishly good service.

IN BRIEF

AngelVPN is a budget provider that punches above its weight in key areas. While it's not quite as feature-rich as Surfshark, or as globe-spanning as ExpressVPN, it's a quality VPN posed at a price that won't break the bank. If you're trying to watch IPTV or streaming services while abroad, you could do a lot worse.

AngelVPN, established in 2022, is headquartered in London. The UK isn't ideal for hosting a VPN company, unfortunately – it's one of the primary members of the Five Eyes data-sharing alliance. This raises questions about how well AngelVPN can deal with law enforcement requests and targeted spying from intelligence agencies.

On a more positive note, AngelVPN has servers in 67 countries, and more than 80 locations, and even boasts coverage across 13 points in America. Plus, in addition to plenty of spots in Europe, such as the UK, France, Italy, and Germany, there are a few lesser-covered regions listed, including Russia, China, Brunei, the British Virgin Islands and the Cayman Islands. This is pretty rare, so if you need a VPN to unblock content in these locations, AngelVPN might be worth a look.

A monthly plan costs \$9.95. This is cheap for a top-tier VPN but no extra features are bundled – you just get the VPN. Going for the six-month plan knocks the price down to \$3.33 per month for a total of \$19.99. Each plan comes with a 30-day money-back guarantee that allows you to test the service before you commit.

Heavenly and hellish

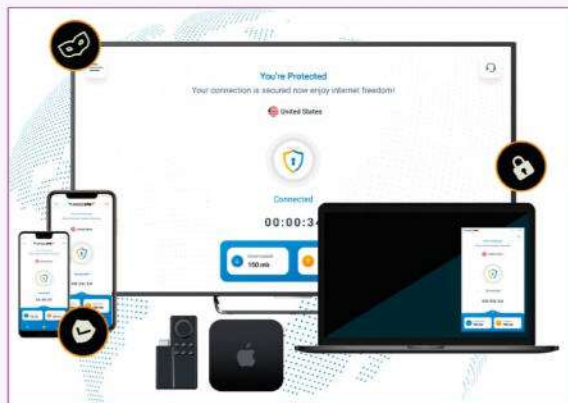
AngelVPN is a mixed bag when it comes to privacy. There's nothing to complain about on a technical level – in fact, AngelVPN is above average in terms of VPN protocol support. However, of the primary concerns encountered with Angel VPN pertains to its privacy policy. Despite claiming to be a strictly no-logs provider, a closer inspection of its privacy policy reveals that it retains several significant data points that could potentially compromise your anonymity. These include details such as the user's city, ISP, connection timestamps, and the specific server being accessed.

While Angel VPN states that it does not record the contents of user connections, the retention of such information could potentially lead to user identification if a third party was able to access the data, especially in the event of a request by law enforcement. Given that AngelVPN is headquartered in the UK, which has a less-than-stellar privacy reputation, it is possible.

AngelVPN works with every streaming site we tested, including Netflix in several regions (America, Japan, Canada and the UK), Disney+, Amazon Prime and Hulu, as well as regional streaming services such as BBC iPlayer, ITVX and All 4.

AngelVPN is good on performance – especially when you consider that it's cheap. To test its speed, we used the London server with a 100Mb/s connection – the test was carried out by Ookla Speedtest.

Connecting over OpenVPN gave us download speeds of 63.85Mb/s and upload speeds of 20.89Mb/s. These are more than enough to connect to a 4K streaming service without buffering or lag – even



■ You'll love their heavenly speeds but perhaps not the privacy features.

with other users on the same connection. The only real complaint is that connecting or changing servers took a bit longer than expected, but your mileage may vary.

AngelVPN is available on all major platforms, such as Windows, Mac OS, Android and iOS. Surprisingly for a smaller provider, there's also a Linux client with a full GUI that works as well as the other desktop clients.

AngelVPN's support is also better than expected. The support staff stole the show – all queries got quick responses via the 24/7 live chat system and email. Each agent was polite and knowledgeable, willing to have an in-depth discussion with us about both the scope of the VPN's streaming support as well as the shortcomings of its privacy policy.

Apparently, AngelVPN prides itself on having highly visible support channels and, as such, maintains an active presence on social media platforms such as Facebook, Twitter (aka X), LinkedIn and Instagram.

The AngelVPN support library is pretty extensive, too, and has quickstart guides for all platform applications. There's even a guide on how you can retrieve your credentials for OpenVPN if you want to set up your own third-party app. **LXF**

VERDICT

DEVELOPER: AngelVPN

WEB: <https://angelvpn.com>

PRICE: From \$2.83 per month for 12 months)

FEATURES	6/10	EASE OF USE	8/10
PERFORMANCE	7/10	VALUE	8/10

Delivers a lot of reliability at prices to suit every budget. It's quick, unblocks plenty of streaming platforms, and works like a charm, but lacks extras and has some logging issues.

» **Rating 7/10**

PureDome VPN

A VPN service that’s aimed at business types like **Mike Williams**, who prefers to do his private business in private.

IN BRIEF

This is based on the consumer service PureVPN, with its large network, decent apps and a good range of features. It’s a relative newcomer to business VPNs, and can’t begin to match the best of the competition for management, access control or reporting tools. If you need fine-tuned user management, detailed audit reports or anything more advanced, we recommend looking elsewhere.

PureDome is a capable business VPN from the people behind PureVPN, one of the most experienced consumer VPN providers around. Signing up gets you all the benefits of PureVPN: easy-to-use Linux, Windows, Mac, Android and iOS apps; thousands of servers across 60-plus countries; WireGuard support for maximum speeds; a kill switch to protect you if the VPN drops; and plenty more.

But PureDome adds business-friendly security, remote access and team management tools of its own. You can control who uses the service and what they can do. Single sign-on support enables your team to access the VPN using their existing credentials, and you can enforce your preferred VPN settings on users for maximum security, and even prevent users accessing the VPN unless, say, they’re running your preferred antivirus.

PureDome is fairly priced for monthly billing at \$8.45 per user billed monthly, but this only drops to a relatively high \$6.74 on the annual plan. There’s also a catch: PureDome has a minimum subscription of five team members, which means the least you can pay is \$42.25 billed monthly, or \$33.70 on the annual plan.

Business VPN apps are often underpowered and have a range of usability issues. That’s no surprise – business VPN providers might know how to create site-to-site network connections through every possible type of firewall, but developing quality apps requires very different skills, and it’s often not a priority.

PureDome is different, because its apps are based on PureVPN’s range. That means a wide choice of platforms, with downloads for Windows, Mac, Android, iOS, Linux, even Chrome support. They include all the features you’d expect from a consumer VPN: IKEv2, WireGuard and OpenVPN protocol support, a kill switch, split tunnelling, location favourites and more – they’re also easy to use, even for the least experienced.

There’s another welcome plus in PureDome’s new support for single sign-on. This enables users to sign in with existing credentials, rather than create yet another account for PureDome. The service only supports Okta and Microsoft Azure AD right now, but hopefully Google and others will land soon.

PureDome makes it unusually easy to give team members access to the VPN. Tap a button, enter an email address, and you can send an invite with a click. Or, if you have a big team, you can import the full list in a CSV file, and send them all at once. As usual, you can assign each user a role that defines what they can do.

PureDome also supports organising users into teams: Sales, Accounts, Engineering, whatever works for your business structure. This enables you to restrict



PureDome VPN supports a full suite of platforms.

each team to its own gateway (Sales can’t access the Accounts systems, say), and you can also apply different VPN access rules to each team. However, we found PureDome couldn’t show us which users were connected to the service, and there’s no connection history, and no information on the devices they use.

Pure speed

We measured PureDome’s performance by running multiple tests on several top speed test sites and apps. We ran our tests from a UK cloud PC with a 1Gb/s connection. The results were very acceptable, with PureDome averaging 610Mb/s on WireGuard. We’ve seen faster VPNs – NordVPN, Surfshark, Hide.me and others reached 950Mb/s-plus – but it’s more than fast enough for most networks, devices and applications.

We checked PureDome’s unblocking performance with eight streaming platforms, using locations around the world. Netflix results were excellent. The service couldn’t get us into Disney+, but it worked with Amazon Prime, and successfully unblocked our other test platforms: BBC iPlayer, ITV and Channel 4 in the UK, and Australia’s 9Now and 10Play. **LXF**

VERDICT

DEVELOPER: PureVPN
WEB: www.puredome.com
PRICE: From \$8.45 per user per month

FEATURES	7/10	EASE OF USE	8/10
PERFORMANCE	8/10	VALUE	7/10

Could work for small teams who need a simple VPN with centralised billing and some quality user and device access controls, but lacking in management and reporting tools.

» **Rating 7/10**

CREDIT: PureVPN



WHATEVER HAPPENED TO SYMBIAN?

The Symbian operating system had a lot going for it, and even dabbled with open source, so why did it fail? **Mike Bedford** investigates.

This part of our computing heritage may now be largely forgotten but, from the mid-'80s, the PDA, or personal digital assistant, had an enthusiastic following. Often thought of as the digital equivalent of the Filofax – a leather-bound loose-leaf paper organiser – these products would be many people's first foray into computing on the move. Their place in history is hardly surprising, given the alternatives available at the time. After all, these were the very early days for laptops, which were bulky and heavy, and had multi-thousand-pound price tags. Smartphones, on the other hand, were still quite some way off.

The first ever PDA, produced by British company Psion, was the Organiser. It appeared in 1984 and had a tiny one-line, text-only monochrome LCD display, and a 36-key keyboard. It provided an electronic diary, searchable address database, calculator and clock. Other applications, including a programming language, were provided as add-on read-only memory modules called datapaks. This handheld device cost £99, many times less than the embryonic laptops of the time. The Organiser was followed by the Psion Series 3, Series 5

and Series 7, which all looked like diminutive laptops. Indeed, they were commonly referred to as palmtops.

Most of these Psion devices used a proprietary operating system called EPOC. The name came from the word epoch, to suggest it was the start of a new era. Not surprisingly for that era, Symbian, as EPOC was later called, was closed source. However, just over 15 years ago, it became an open source OS. This was also around the time that Android was emerging from the shadows. And, as we'll see later, this was ultimately one of the causes of Symbian's demise. But there's a cautionary tale here, because the story of Symbian shows that it would be wrong to always take the view 'once open source, always open source'. Intrigued? Well, do read on, and all will be revealed.

The beginning of time...

To go back to Symbian's roots, we need to delve into EPOC. Developed by Psion and released in 1989, it first appeared in the company's MC 200 and MC 400 products, which were laptops, rather than the company's more familiar PDAs. This 16-bit OS was designed for Intel x86 compatible processors, and it

also ran on Psion's first palmtop PDA, the Series 3. But things were changing, and this brings us to Symbian.

Following the huge success of the Series 3 – which sold 1.5 million units – Psion started its migration to 32-bit computing with the Series 5, which launched in 1997. But while PC manufacturers tended to stick with the x86 architecture when they moved to 32 bits, Psion took a different approach. Its processor of choice was an ARM7 variant, a move that predated the introduction of ARM chips into smartphones, in which that architecture is now dominant.

With a new version of EPOC, EPOC32, Psion had new aspirations for its 32-bit OS. With the aim of capitalising on the growth of the mobile computing market, which would soon morph into smartphones, Psion launched Symbian Ltd in 1998. Initially a partnership between Psion, Nokia and Ericsson, later members included Motorola, Matsushita, Siemens and Samsung. What's more, Symbian OS – as EPOC32 became soon after the change in ownership – was also licensed to Arima, BenQ, Fujitsu, Levona, Sharp and Sony. It's not hard to appreciate, therefore, how influential it was in the smartphone arena in those early days. Indeed, in 2006, it claimed 67% market share.

Going deeper

We'll come back to the Symbian timeline in due course. However, in the light of our theme of 'whatever happened to Symbian?' and its eventual demise, we need to poke a bit deeper into the OS. Symbian was split into two layers: the kernel, which interfaced with the hardware and provided the basic functionality, and the user interface. Nothing too unusual in that, you might think. But where it differed from Android, for example, is that this had a far more profound effect on app developers, and hence on users.

First we should point out that Symbian had several user interfaces, of which the main ones were S60, UIQ and MOAP. The existence of multiple user interfaces – including the command line – won't seem at all unusual to most Linux users. But there's a difference, and that difference had negative consequences. Linux users are a mixed bunch, and often experiment with different options – user interfaces included – before homing in on a preferred solution. Most smartphone users, on the other hand, just want something that works, straight out of the box. Even so, surely variety is the spice of life in the area of user interfaces. Well, yes and no, because the choice of interface wasn't available to the user, but based on a decision made by the phone manufacturer. But unlike the situation with Linux, software developed for one user interface wouldn't work on another. So, for example, software written for S60 wouldn't run at all on UIQ or MOAP. What's more, it has been suggested that releasing an app for two different Symbian user interfaces was almost as big a job as releasing it for two different operating systems, say iOS and Android. For a developer, having to port Symbian applications from one user interface to another wasn't an attractive option, and this must surely have had a negative impact on software availability; indeed, we'll see some figures later. In turn, we can imagine that this was one of the factors in Symbian's eventual downfall.

And there's more, again concerning the attractiveness, or otherwise, of Symbian to the

application developer. Reportedly, it was a swine from a programming perspective. Speaking to *Wired* in 2010, Freddie Gjertsen, head of product development at London-based app developer Touchnote, gave some idea of the uphill task of writing code for Symbian OS. "For us to build a plugin that connects the camera to the gallery of photos in the application took about four to five weeks of work. In Android, it took us five minutes. It was a feature built into the OS and we had to just turn it on." And Gjertsen, we might add, had previously worked with Symbian for five years. So, just what was so difficult about Symbian program development?

It rather seems, from Gjertsen's comment, that Symbian often required app developers to write code from scratch that was already built into the operating system in Android. But other factors have been cited on several occasions. Another drawback concerned the programming languages available for app developers. In time, a range of languages would become available but, arguably, the damage was already done because, in the early days, choices were limited. In fact, it wouldn't be too far short of the mark



Symbian's rise to supremacy in the smartphone market started with the Nokia 7650 in 2002.



EPOC, later to be called Symbian, first appeared on Psion's highly successful Series 3 palmtop computer, otherwise known as a PDA or organiser.

» PSION SERIES 3

Symbian's final years were mostly spent empowering smartphones from various manufacturers, a market in which it excelled prior to its dramatic downfall. But, since the form factor has now largely disappeared, we thought it would be interesting to look at one of Psion's most successful organisers, the Series 3, which is where Symbian, then called EPOC, first appeared.

In presenting historic computers here at *Linux Format*, we've often found machines with almost unimaginably basic hardware specifications. So we've been talking of kilobytes where, today, we'd expect to see terabytes of memory, to give just one example. The Psion Series 3 might not be as ancient, having hit the market in 1991, but its spec might still be an eye-opener. It had an NEC V30, 16-bit x86 compatible processor, running at 4.7MHz. The built-in software was housed in a 1MB ROM, and there was 256KB of RAM. And the monochrome screen had a resolution of 240x80, although it had a small screen, making the low pixel resolution not so noticeable.

But while, in other respects, its small screen would be a disadvantage, the Series 3's whole rationale was based around its small dimensions. It measured 165x85x22mm and weighed just 265g. Best of all, though, it was powered by two AA batteries, which would last an incredible – even now – 20-35 hours.

to say that early Symbian app development depended on C++. This language has never been considered an easy one to learn; indeed, it scores well down in this respect compared to most of today's other commonly used languages. However, in the early days of Symbian, C++ hadn't become standardised. So, the language associated with Symbian was a proprietary version of the language. And the general consensus was that it was notoriously difficult to learn and to use.

Taking all these points together, the findings of a global survey of more than 400 mobile app developers, published in 2010, probably isn't too surprising. Mastering Symbian takes 15 months, it found, compared to less than six months for Android. And the upshot of this wasn't hard to see. The good news is that, by the time Apple launched the iPhone, there were 10,000 apps available for the Symbian platform. However, there's a big 'but'. It had taken over seven years for Symbian to reach that figure, while Apple was



Psion's first product, the Organiser, didn't even have an operating system, but it led to much greater things.

» THE LINUX OPTION

Today the idea sounds strange – even though we acknowledge that Android has a Linux kernel deep within – but there was a time when Linux was a serious contender for smartphones. In 2005, Nokia launched Maemo as an open source OS for smartphones and tablets. It was a modified version of the Debian GNU/Linux distro. It was bundled with the expected selection of apps, and more could be installed via an Application Manager, or APT. That's right, Maemo-based devices offered users the option of using a command-line interface. Actually, most such devices were tablets, and it seems the only smartphone to embrace Maemo was the Nokia N900. And as part of the breaking down the dividing line between the smartphone and the PC, we should point out that the N900 also had a physical keyboard via which you could enter your textual commands.

So, what happened to Maemo? In 2010 Nokia teamed up with Intel to produce another OS. Called MeeGo, it was created by merging Maemo with Intel's Moblin (short for Mobile Linux). The companies' aspirations went beyond notebooks and smartphones to encompass other consumer devices, including in-car entertainment systems and smart TVs, but that didn't assure its future. Its final release was in 2012, by which time it had only appeared in a couple of smartphones.



Nokia's N900 was one of the few smartphones to be powered by Linux, in the form of Maemo. It even had a slide-out keyboard and the option of a command-line interface.

able to boast 10 times as many just over a year from launching its first SDK for iOS.

The open source experiment

Returning to the Symbian timeline, we find a move that was, perhaps, rather surprising, but one that was surely no less welcome. That move was the acquisition of Symbian by Nokia. But while that doesn't sound too unusual, the same isn't true of what Nokia did with the operating system back in the early 2000s. In particular, Nokia set up an independent non-profit organisation called the Symbian Foundation. This, in turn, paved the way for the core OS and its user interfaces to become free open source software. The Symbian Foundation was established in April 2009, and the source code published in February 2010. According to the Symbian Foundation, this was the largest ever migration of code to open source in the history of software. That move by Nokia was surely made for commercial reasons, rather than out a sense of generosity, but open source is still open source. And it did seem to go well for Nokia, with the launch of its N8 – in April 2010 – which, apparently, benefited from the first release of a new open source Symbian, namely Symbian^3. According to Lee Williams, executive director at the Symbian Foundation, "We are stoked that Nokia has chosen our fully open source S^3 for their latest device, the Nokia N8. S^3 enables an unparalleled set of options for device creators and app developers to extend the usefulness of Symbian products and services and, in turn, enhance the lives of mobile consumers. This is an example of what is possible with the latest and greatest version of the Symbian platform." What could possibly go wrong?

According to many pundits of the era, it was already too late for Symbian OS when it was released as open source software. And, if so, as many suggested, all it could have been expected to do was extend its life. In other words, although the writing might already have been on the wall for Symbian OS, the open source initiative would motivate a new pool of developers to work on the OS, keeping it going while Nokia concentrated on its next-generation OS. In fact, Nokia considered several such replacements, and you can read about two of them, the Linux-based Maemo and MeeGo, in the box (left). But those two operating

systems would become little more than a footnote in the history books. After all, iOS and Android were already stealing market share from Symbian OS, something that's hardly surprising when we consider they had the might of Apple and Google behind them. And the figures were startling, especially if we bear in mind Symbian's 67% smartphone market share in 2006. By the third quarter of 2010, this stood at 39%, dropping to 31% by the fourth quarter. Furthermore, it fell behind Android for the first time that same quarter.

Symbian bites the dust

The future of Symbian was predicted – or at least recommended – less than 12 months after its migration to open source, by Nick Jones, vice president and analyst with technology research and consultancy firm Gartner. Writing on his blog, Jones stated, “The brave Symbian open source experiment has failed. The only two top-tier device manufacturers on the Symbian board other than Nokia have deserted it.” Indeed, Jones's comparison of the Symbian Foundation's management of the operating system with rearranging deckchairs on the Titanic was widely reported. And it gets worse, as Jones followed this up with the recommendation that “what Symbian needs is agility and vision, not committees, and if Symbian is fixable it will be fixed a lot faster under a single leader. And great user interfaces aren't developed by committees.” Reportedly, he also said that “Nokia needs to scrap the Symbian Foundation and bring the operating system home”, a statement that proved to be prophetic. And this brings us to November 2010.

About 18 months after the Symbian Foundation was established and just nine months after the release of the Symbian code into the public domain, Nokia announced that it would take back the development of Symbian, and the Symbian Foundation would become nothing more than a licensing organisation. And shortly afterwards, all source code and associated material



It might have been the last ever Symbian smartphone, but the Nokia 808 PureView ensured it went out with a bang.

was removed from the Foundation's website. Nokia's foray into free open source software had ended.

So, what became of Symbian following its return to the Nokia fold? First of all, as already mentioned in passing, we should point out that Nokia, now lukewarm about Symbian, was dabbling with several alternative operating systems. Included here were the Linux-based Maemo and MeeGo, and Microsoft's Windows Phone. It might be hardly surprising to learn, therefore, that Nokia's last ever Symbian phone was its 808 PureView, which hit the streets in May 2012. Indeed, just eight months later, Nokia confirmed that the 808 PureView would be its last phone to use Symbian. And it rather appears that there were no further Symbian phones from other manufacturers either. In bowing out with this model, though, Symbian didn't die out with a whimper. The 808 PureView won Best New Mobile Handset, Device or Tablet award at Mobile World Congress 2012, and the award for Best Imaging Innovation for 2012 from the Technical Image Press Association. Surely there could have been no better swansong for Symbian. **LXF**

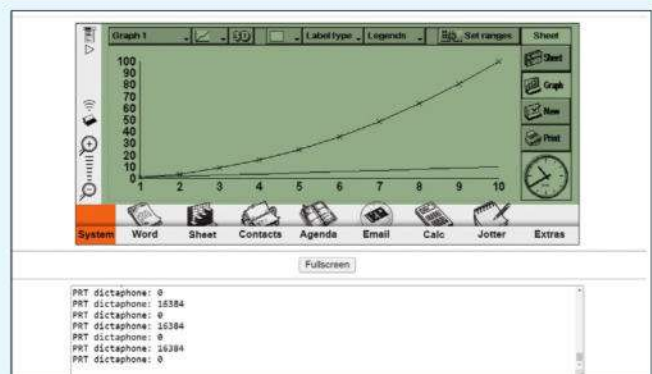
» EMULATING THE PSION SERIES 5

If you'd like to know what it was like to use a Psion organiser of old, you could buy one second-hand on eBay or similar. They don't cost a fortune if you shop around and don't insist on having one in pristine condition. But if your house is already overflowing with clutter, how about getting a feel for one via emulation?

We note that the MAME emulator, which is available from Linux, supports several Psion organisers. However, it requires you to find the relevant ROM image – the software provided on the organiser – for yourself. Since that software is

proprietary, we won't suggest you undertake something that might be illegal. However, we did find an online emulator at <https://wuffs.org/WindEmu/> that you might like to try.

There's probably not a lot we need to tell you since it's mostly fairly intuitive, but here's what we discovered. First of all, it's not fast to start up. Only when text in the box below the emulator reduces to showing only lines starting 'PRT dictaphone:' are you ready to go. This might take up to a minute, although parts might work before that. However, the biggest issue is typing



Fancy getting a feel for 27-year-old computing on the move? This online emulator lets you get to grips with the Psion Series 5mx.

characters such as mathematical operators, which shared a key with another character. These required the use of a function

key on the Psion, which appears to be the Windows/Super key in the emulator on a PC. We didn't find them all, though.

tom's **HARDWARE**

GET ALL THE ESSENTIAL BREAKING NEWS FOR THE TECH ENTHUSIAST!

No matter if you're building a PC, buying a laptop or learning about robots, Tom's Hardware has all the comprehensive knowledge you need.

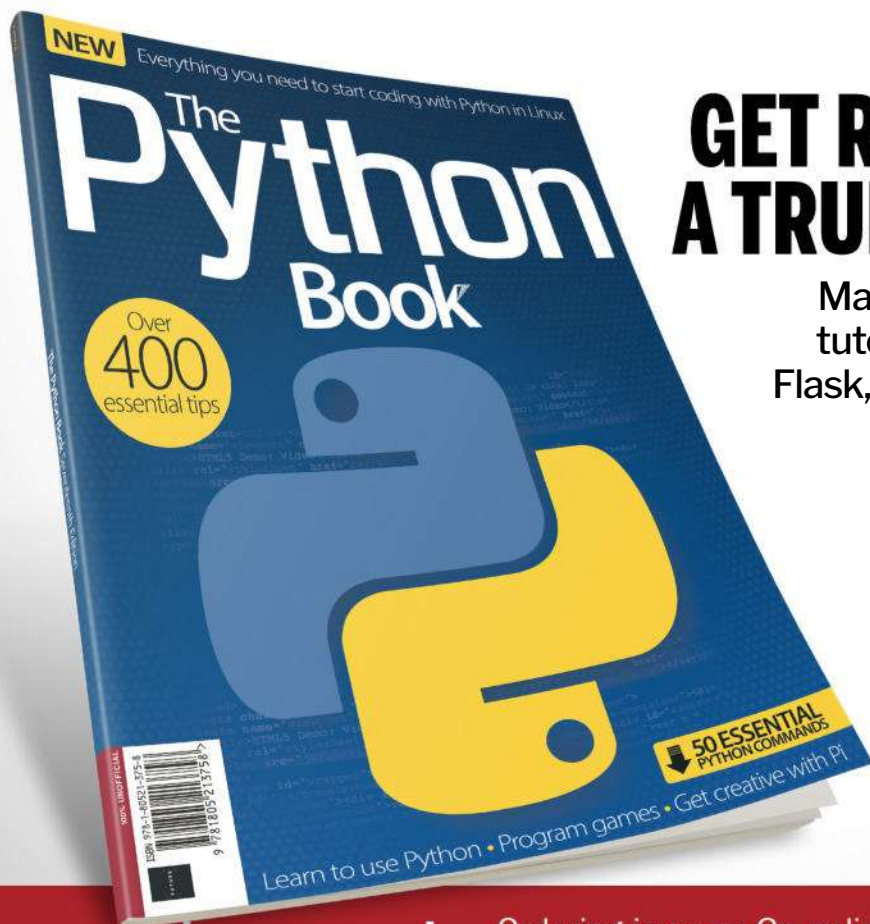


**Scan & Subscribe
for free!**



GET READY TO BECOME A TRUE PYTHON EXPERT

Make Python work for you with tutorials on coding with Django, Flask, Pygame and even more useful third-party frameworks.



**ON SALE
NOW**

Ordering is easy. Go online at:
magazinesdirect.com
Or get it from selected supermarkets & newsagents

HotPicks



Mayank Sharma

is grumpier than usual since he binge-watched *TDF: Unchained*, and was then a domestic to open source gems as he raced to the deadline.

XFE » Tauon Music Box » Puddletag » Scrot » Dysk » Web Apps » Gear Lever » RetroArch » The AMC Squad » LibreWolf » HP Linux Imaging and Printing

FILE MANAGER

XFE

Version: 1.46.2

Web: <http://roland65.free.fr/xfe/>

If you aren't using Linux on a well-endowed machine, you can scrape off some load from the hardware using the *X File Explorer (XFE)*. This lightweight file manager is an interesting mix of the classic interface with modern day flexibility. The fact that it can be operated from the keyboard alone will please experienced campaigners.

The *XFE* project doesn't put out any binaries, but it isn't too cumbersome to compile it from source. Grab the compressed tarball from its website and extract it with `tar -xvf xfe-*.tar.xz`. Then run the `./configure` script, and use your distro's package manager to install any missing dependencies, if at all. Then type `make`, followed by `make install`.

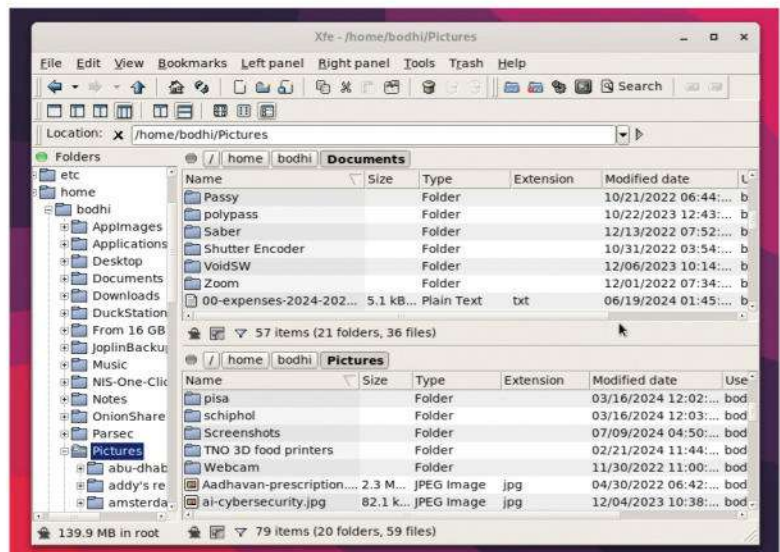
If that sounds too cumbersome, you can also pull in *XFE* from your distro's official repos. Remember, though, that the version in your repos might be behind by a dot version or two. Fedora users can install it with `sudo dnf install xfe`, while `sudo apt install xfe` fetches it for Debian and Ubuntu users.

At first glance, you might be turned off by *XFE*'s old-school appearance. But unlike your distro's default file manager, *XFE* comes up in a jiffy, and it does so without cutting any of the useful file manager functionalities.

At the top of the window, *XFE* sports a menu bar followed by several toolbars. None of these toolbars are glued to their position, and you can move them around. Of course, you can also head to the View menu and hide any or all of them.

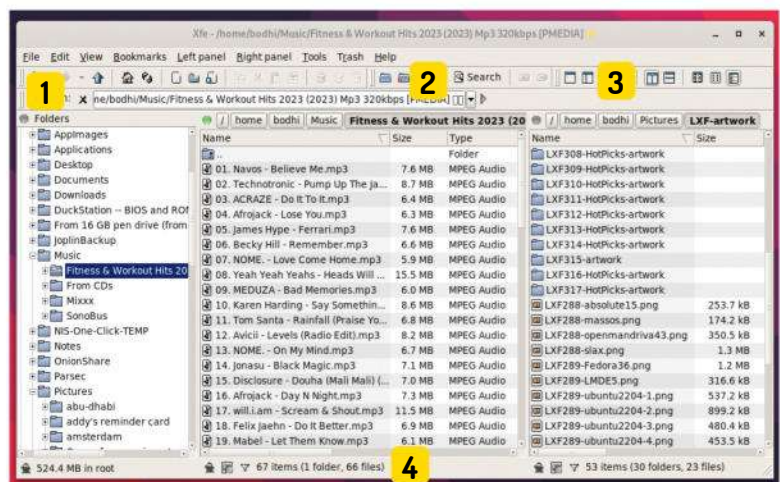
Beneath these, *XFE* has a location bar and a multipanel interface. On the left you get a directory tree, while on the right is the more familiar directory listing. Just like the toolbars, you can hide several other components in the file manager, such as the side panel.

Some of *XFE*'s most useful features are its fleshed-out context-aware right-click menu, and a very thorough find utility.



XFE has four prebaked layouts. By default, it uses the tree and panel, but it can also display tree and two panels, stacked next to each other or horizontally.

LET'S EXPLORE XFE...



1 General toolbar

Here you get buttons for the standard file manager actions. You can do all of these and more from the menu bar, so you can hide this toolbar if you don't use these functions often.

2 Tools toolbar

This toolbar houses some very useful functions. You can, for instance, launch a new normal *XFE* instance, or one with root privileges. There's also a button to launch *XFE*'s find utility.

3 Panel toolbar

Here you'll find buttons to change the appearance of the file manager. You can split the *XFE* window into two panes. Two panes with tree view, one pane with tree view, or a single pane.

4 Status bar

This area displays different information in the tree view and in the panels. In the tree view, it displays the size of the current folder, while in the panel it displays a plethora of useful details.

MUSIC PLAYER

Tauon Music Box

Version: 7.8.0

Web: <https://tauonmusicbox.rocks>

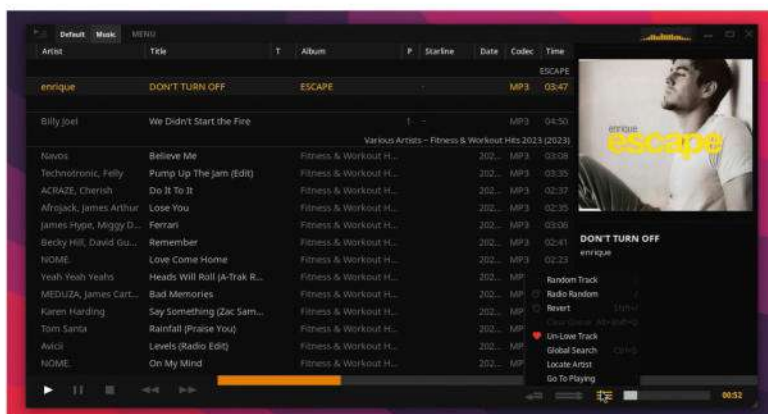
There's no shortage of apps that play MP3s. Heck, you don't even need a dedicated app for this, as many modern file managers ship with the functionality built-in. But if you want more control over your music playback, you have to use a dedicated music player, such as *Tauon*.

Tauon is officially distributed on Flathub, and can be installed with `flatpak install flathub com.github.taiko2k.tauonmb`.

Unlike its name, the app is fairly straightforward. *Tauon* looks stylish with its all-black tabbed interface, and the app can play various types of local audio files including MP3s, FLACs and OGGs.

On first launch, it's best to head to Menu > Import Music Folder to let the app recursively grab all tracks housed under your `~/Music` folder.

Use Alt+T to create a new playlist, which opens in a new tab. You can then populate it by dragging and



dropping tracks from the file manager. To name the playlist, right-click on the tab. While you are there, take a look at the Sort submenu, which houses a handful of options to rearrange the tracks in your playlist.

The menu also has a Generate submenu to create a new playlist automatically. It offers over a dozen useful options, such as Top Played Tracks, Top Rated Tracks, Longest Albums and lots more. Once you've created your playlist, double-click on a tab to play its songs.

Tauon can also display lyrics for the playing track. You first have to enable one of its supported services by heading to Menu > Settings > Function > Lyrics. After you've enabled the sources, right-click on the cover art of the currently playing song and select the Search For Lyrics option. *Tauon* then searches for the lyrics on the enabled services and displays them.

Tauon has interesting theming options. Head to Menu > Settings > Themes to switch to a different colour theme. The app can also use album art as a background.

AUDIO TAG EDITOR

Puddletag

Version: 2.4.0

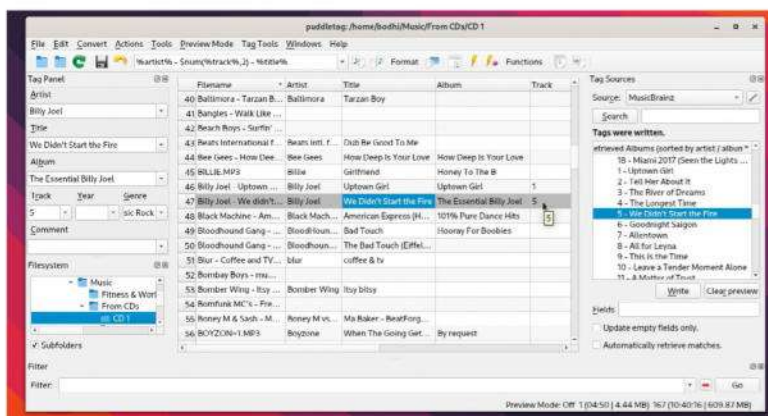
Web: <https://docs.puddletag.net>

Any music player, including *Tauon*, needs the music files to have proper tags to work their magic. The players use audio tags to display details about the track, such as its artist, album name, year of release and such. They also use these tags to fetch relevant artwork, lyrics for the track, details about the artist, and a lot more.

If your audio tracks are missing these details, you can key them in with a tag editor, such as *Puddletag*. The app is available in the official repos of popular distros such as Debian, Fedora and Ubuntu, but the best way to install it is to grab it with Python's package manager, *PyPI*.

Before you can use *PyPI*, first install Python 3 from your distro's package manager with `sudo apt install python3` for Ubuntu, or `sudo dnf install python3` for Fedora. You can now use *PyPI* to install *Puddletag* with `pip install puddletag`.

The app has a rather busy interface that looks too cumbersome for a straightforward task. Its developers justify its appearance saying the app



uses a spreadsheet-like layout, so that all the tags you want to edit are visible and easily editable.

Go to File > Open Folder to point the app to your music repo. It then imports all tracks and displays all their tags much like a spreadsheet. The app supports all the popular tag formats including ID3v2 (for MP3s), VorbisComments (for OGG), and more.

You can now add and edit the tags manually by double-clicking any cell of the selected track. Usually, filenames have enough details about a track, such as the name of the artist, the track number, and the title of the song. *Puddletag* can extract these details from the title and append them in the relevant field with the help of patterns. The project's website has detailed documentation on how to use patterns.

With *Puddletag*, you can tag audio files using metadata from external tag sources, such as Musicbrainz and FreeDB, and cover art from Amazon.

SCREENSHOT UTILITY

Scrot

Version: 1.11.1 Web: <https://github.com/resurrecting-open-source-projects/scrot>

Even if you discount the apps that ship with virtually all desktop environments, Linux has some wonderful standalone screenshot tools, such as *Shutter* (LXF317, page 88). But what if you're running Linux on an underpowered machine? It just about works with your customised desktop running the *i3* tiling manager.

If you're running such a rig (or you just can't imagine life outside of a terminal), you'll love *Scrot*, a CLI screenshot utility. You can compile the utility from source, but it's a lot easier to get it from your distro's official repos. That's `sudo apt install scrot` for those running Ubuntu, and `sudo dnf install scrot` for Fedora users.

To capture a screenshot, crack open a terminal window and type `scrot [filename]`. If you don't include a name for the file, *Scrot* creates one for you, along the lines of `2024-07-10-041751_1366x768_scrot.png`.

Running *Scrot* with no options takes a screen capture of your entire desktop. To screenshot a

```
[bodhi@fedora ~]$ scrot
[bodhi@fedora ~]$ scrot screen.png
[bodhi@fedora ~]$ scrot -u window.png
[bodhi@fedora ~]$ scrot -u -s window.png
[bodhi@fedora ~]$ scrot -u -d 10 window.png
[bodhi@fedora ~]$ scrot -u -s -d 10 window.png
[bodhi@fedora ~]$ scrot -u -d 15 -c
Taking shot in 15.. 14.. 13.. 12.. 11.. 10.. 9.. 8.. 7.. 6.. 5.. 4.. 3.. 2.. 1..
0.
[bodhi@fedora ~]$
```

single window, use `scrot -u`. This tells *Scrot* to grab the window that is currently in focus. This just captures the terminal window you're working in. Another *Scrot* peculiarity is that it always captures a little bit more than the window.

You can take care of both with the `-s` option. This option can do one of two things. You can either select an open window or draw around the window or the portion of the screen/window you want to capture.

However, that captures everything in the foreground as well. A better option is to delay the capture to hide all windows except for the one you want to capture. You can do this with `scrot -u -d [num]`. The `-d` option tells *Scrot* to wait before grabbing the window, and `[num]` is the number of seconds to wait.

Slap on the `-c` option to create a countdown in your terminal when you use the `-d [num]` option. It's kind of pointless, since the terminal window would be the first you'd minimise.

FILESYSTEM LISTING UTILITY

Dysk

Version: 2.9.0

Web: <https://dystroy.org/dysk/>

If you need information about your disk, you need to fire up a terminal and use the `df` command. It does a good enough job, especially when used with the `-h` option. However, *Dysk* does the same job, only better.

Dysk releases binaries that you can grab from the Download section of its website. The x86-64 Linux build works for virtually all users, but on top of this, there are also releases for ARM32, ARM64, OpenWRT and other hardware.

Once you have the *Dysk* binary, you have to first give it executable permission with `chmod +x dysk`. You can then use it from the directory it's housed in. A sensible option is to move it to `/usr/local/bin/` so that you can use it from anywhere.

To get a basic overview of your internal and any connected disks, type `dysk`. It tells you the location of the device, the type of filesystem, the kind of disk, the space used, both in absolute space and in percentages, and how much is free. And a nice little graphic does a good job of conveying the space details.

```
[bodhi@fedora ~]$ dysk -c +inodes+label
filesystem type disk used use free size mount point inodes label
/dev/sda6 ext4 HDD 230G 81% 52G 283G / 6% 
/dev/sdb1 exfat remov 6.3G 40% 9.2G 16G /run/media/bodhi/Ventoy 
/dev/sdc1 vfat remov 309M 16% 1.7G 2.0G /run/media/bodhi/2GB 2GB

[bodhi@fedora ~]$ dysk -af 'type=squashfs & used > 400M'
filesystem type disk used use free size mount point
/dev/loop8 squashfs HDD 530M 100% 0 530M /var/lib/napd/snap/gnome-42-2204/176
/dev/loop11 squashfs HDD 471M 100% 0 471M /var/lib/napd/snap/kf5-5-105-qt-5-15-9-core22/11
/dev/loop12 squashfs HDD 460M 100% 0 460M /var/lib/napd/snap/kf5-5-110-qt-5-15-11-core22/3

[bodhi@fedora ~]$
```

The Disk column tells you whether it's a removable device (remov), HDD or the traditional platter-based hard disk, an SSD, RAM and so on. It can even identify LVM partitions, and encrypted ones.

You can add more information about your disks and tailor the displayed columns to your own particular needs and preferences with the `-c` option. For instance, `dysk -c label+inodes+` adds two columns to the output that show the label of the filesystem and another graphical representation of the number of inodes used and available.

You can adjust the column order as you see fit – for instance, `dysk -c +inodes` adds the inodes column at the end of the existing columns.

Another neat trick is the ability to filter things out. For instance, `dysk -af 'type=squashfs & used > 400M'` displays SquashFS types that use more than 100MB of storage.

APP CREATOR

Web Apps

Version: 0.5.1 Web: <https://codeberg.org/eyekay/webapps>

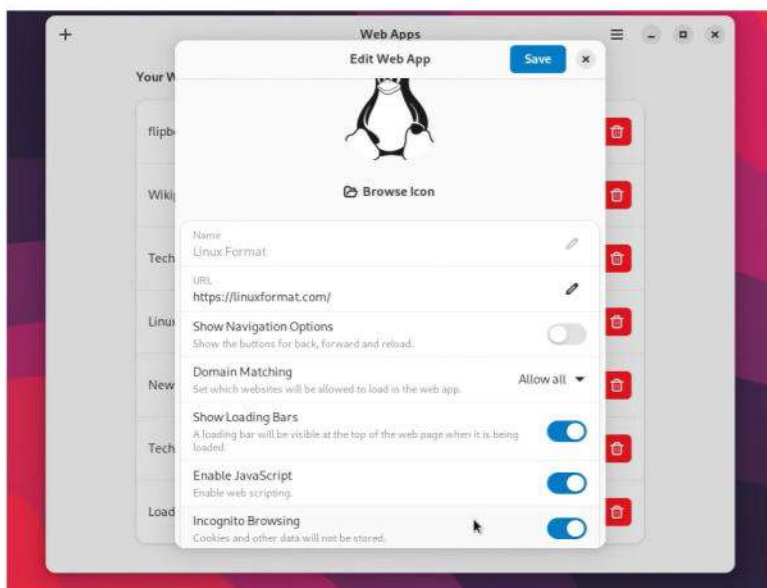
To head to a website, you first have to launch a browser, then key in the website address. If there are certain websites you visit daily, you can save yourself some time by rolling those sites into an app. You can then launch it from your distro's app launcher, which directly takes you to the website. That's exactly what *Web Apps* is designed to do.

It's distributed on Flathub, and can be installed with `flatpak install flathub net.codelogistics.webapps`.

The app has a straightforward interface. To create an app from a website, press the + button and enter the URL. This can be its landing page (such as **techradar.com**) or a section (such as **techradar.com/pro**).

Web Apps then analyses the website and presents an Add Web App dialog box. It fetches the icon for the website by itself, but you still have the option to use an icon of your own.

Similarly, it also puts in the name of the website, which becomes the name of the app it creates. While it's usually best to just stick with the default name, you can edit it if you want.



Besides the icon, name and website address, *Web Apps* also has a few settings to control the behaviour of the app. To begin with, you can go with the default options, since you can bring up these options even after you've rolled a website into an app.

By default, *Web Apps* enables JavaScript, and shows loading bars in the app. Optionally, you can also ask it to show the back, forward and reload buttons. Without adding these buttons, you can't use any of these functions, even through their keyboard shortcuts.

You can also ask *Web Apps* to open a website in incognito mode. When toggled, this blocks the website from storing cookies and other data.

APPIIMAGE MANAGER

Gear Lever

Version: 2.0.0 Web: <https://mijorus.it/projects/gearlever>

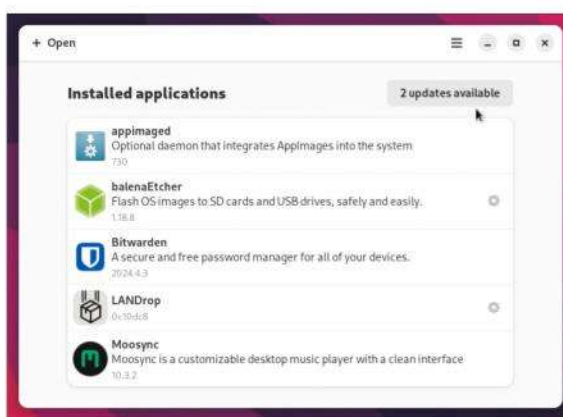
Most projects now release their wares in their preferred distro-agnostic format for distributing portable apps, namely AppImage, Snaps or Flatpak. They all have their pros and cons.

The self-contained AppImages involve no installation as such; give the downloaded AppImage executable permission, and you're good to go. But it's a different matter to add it to a distro's application launcher. They also don't have a unified console for administering them.

This is exactly what you get with *Gear Lever*. You can use it to manage all the AppImages in your system, which *Gear Lever* also then adds to your applications menu.

Amusingly, *Gear Lever* is distributed as a Flatpak, and can be installed with `flatpak install flathub it.mijorus.gearlever`. If you already have the app (we've featured it earlier in **LXF306**, page 82), you can update it with `flatpak update it.mijorus.gearlever`.

Gear Lever has a very approachable interface. You can drag and drop an AppImage from the ~/Downloads



folder, or wherever you house your AppImages. You can also use the + icon to browse your filesystem and point the app to an AppImage.

All apps added to the application menu are listed in *Gear Lever*'s main interface. You can now use *Gear Lever* to update all your AppImages. To do so, bring up the app's preferences and add the URL to that particular AppImage's downloads page. Repeat for all AppImages you want to manage through *Gear Lever*.

To check if any of the AppImages have an update, hit the Updates button in the main interface. The app displays the number of AppImages with updates available and marks them with an updates icon.

You can then click on the app to head to its preferences, and press the Update button to grab its latest release.

Although you have to set it up manually, there's no beating the convenience of updating AppImages using *Gear Lever*.

RETRO GAMING

RetroArch

Version: 1.19.1**Web:** www.retroarch.com

If you like classic games (who doesn't?), you can run them through *RetroArch*. It's a unified front-end for all kinds of emulators. *RetroArch* has a PPA for Ubuntu and is also available as a Snap, AppImage and Flatpak. You can install it from Flathub with `flatpak install flathub org.libretro.RetroArch`.

RetroArch has a cumbersome-looking interface with a whole lot of menus and submenus. Before you load your ROMs, you need to load an emulator, or core. Head to Main Menu > Load Core > Download A Core.

RetroArch displays a long list of supported cores. Many have multiple cores. This is good because your ROMs might not be compatible with all cores. Browse the list and click on a core to download it. Repeat the process to download other cores as well.

You can now dive in and play a game, but it's best to first make *RetroArch* aware of your ROM collection. To do this, head to Import Content > Scan Directory. Now navigate to the directory that houses your ROMs and click the Scan This Directory button.



Once it's done, find your way back to the main menu. Here you'll notice new entries at the end of your menu. For instance, we had ROMs for the Sony Saturn, and after importing the contents of our game directory, we got an entry named Saturn.

Click on the core to list all imported games, and click on a game to play it. You can press F1 to pause the game, which brings up the in-game menu. This has several interesting options. For instance, you can record gameplay, and even stream it if you've set it up. There's also the On-Screen Overlay option that dresses up the black borders around the game.

Along with an endless list of retro consoles, *RetroArch* also works with a wide range of controllers right out of the box.

FPS

The AMC Squad

Version: 4.5.2**Web:** <https://amcsquad.itch.io/game>

On the subject of retro gaming, here's a game that does a nice job of blending the old with the new. *The AMC Squad* bills itself as a retro FPS with modern sensibilities. It actually started out as a mod for *Duke Nukem 3D*, before evolving into a standalone game.

You can download it from its website and follow the handful of instructions to build the game. Or, as it suggests, just grab it from Flathub with `flatpak install flathub io.itch.amcsquad.amcsquad`.

The basic premise of the game is that the Earth Defence Forces assembles the AMC Squad to defend humanity after the demise of its greatest ever hero, Duke Nukem.

You need to select one of the eight playable characters, each with their own abilities, and a set of weapons, from swords and guns to magical spells. The



The AMC Squad's gameplay is the real kicker. The story gives the shooting and running a much-needed purpose.

game currently features four episodes, each one spanning several hours of gameplay.

The first episode came out in 2011, the second in 2015, the third in 2019, and the fourth in 2022. While they work on episode five, the developers have put out an episode 4.5 release with several new features.

The game lets you play any chapter. With five difficulty levels, the first is designed to explore the story instead of engaging the enemy. Newbies are encouraged to get their bearings in a training map.

Playing the game isn't any different from any other FPS. Head to Options to tweak the game settings. You can review and adjust the keyboard, mouse and controller behaviour, and also fiddle around with the display and sound settings. But none of this is necessary as the game ships with adequate defaults. The game also has an autosave feature that's really handy to resume campaigns.

WEB BROWSER

LibreWolf

Version: 127.0.2-1.2

Web: <https://librewolf.net>

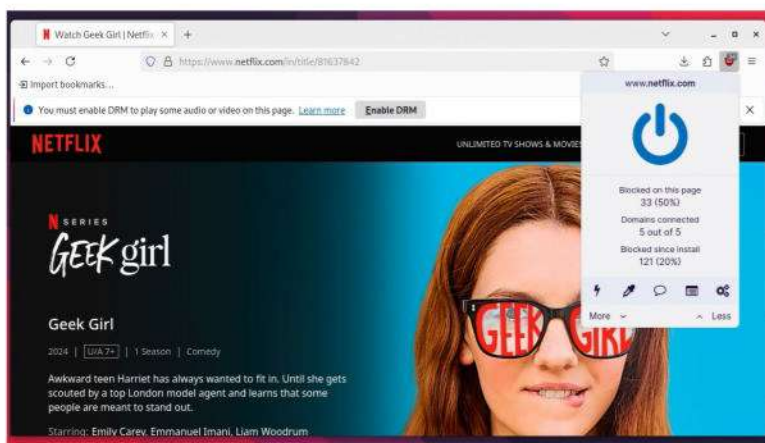
If you're concerned about privacy, chances are you aren't using the regular popular browsers. Instead, you're probably using something based on the popular browsers, but tuned to help safeguard your privacy. *LibreWolf* is one such browser.

LibreWolf hosts repos for all the popular desktop distros including Ubuntu, Fedora, OpenSUSE and more, along with installation guides for each. You can also install it from Flathub with `flatpak install flathub io.gitlab.librewolf-community`.

The browser is a customised version of *Firefox* and looks the part. But while it looks and feels like any other browser, the web it presents looks very different. Media doesn't play automatically, and you don't get any search suggestions or extension recommendations.

Unlike *Firefox*, *LibreWolf* defaults to DuckDuckGo. You can head to Settings > Search to replace this with other privacy-respecting search engines, such as StartPage, SearXNG, MetaGer and others.

Most of the differences between *Firefox* and *LibreWolf* aren't visible. Along with privacy, *LibreWolf*



pays attention to security, so it enables HTTPS-only mode by default and tracks upstream *Firefox* releases to ensure it has the latest security fixes.

It also zaps all temporary files when you exit the browser, and always prompts you for a filesystem location when you download something. *LibreWolf* also tweaks several default *Firefox* features that can be abused by unscrupulous sites. For instance, it disables *Firefox*'s telemetry data collection, search and form history, form autofill and more.

One of the noticeable differences is the inclusion of the uBlock Origin plugin, which helps the browser achieve its objectives. It ships with custom filter lists, and is set to strict mode, which helps it block trackers and more.

While *LibreWolf* works with most websites, there are some that will break because of its hardened stance against DRM and cookies.

PRINTER MANAGEMENT TOOL

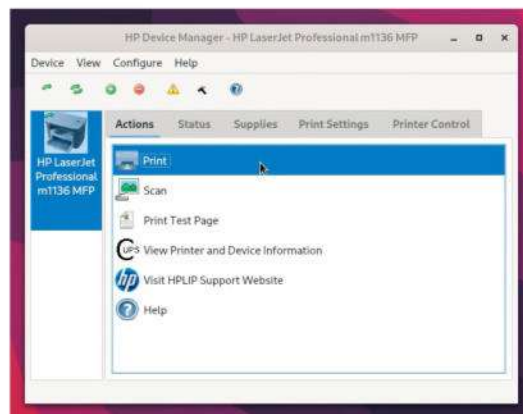
HP Linux Imaging and Printing

Version: 3.24.4 Web: <https://developers.hp.com/hp-linux-imaging-and-printing>

If you have an HP printer or multifunction device, there's no better option than *HP Linux Imaging and Printing (hplip)* to control it. It supports over 2,000 printers from HP and virtually all popular desktop distros.

Head to the utility's homepage to download the latest version of the utility (currently `hplip-3.24.4.run`) for your distro. Before you run the script, with `sh hplip-3.24.4.run`, make sure your printer is switched on and connected. It's best to choose the automatic installation mode and go with the default options, marked with *. The script detects your distro and also installs any missing dependencies.

When it's done, you'll have the *HP Print* utility in your applications menu. The utility has a simple two-pane interface. On the left it lists all your configured HP printers; on the right you have a tabbed interface that bundles all kinds of tools for your printer.

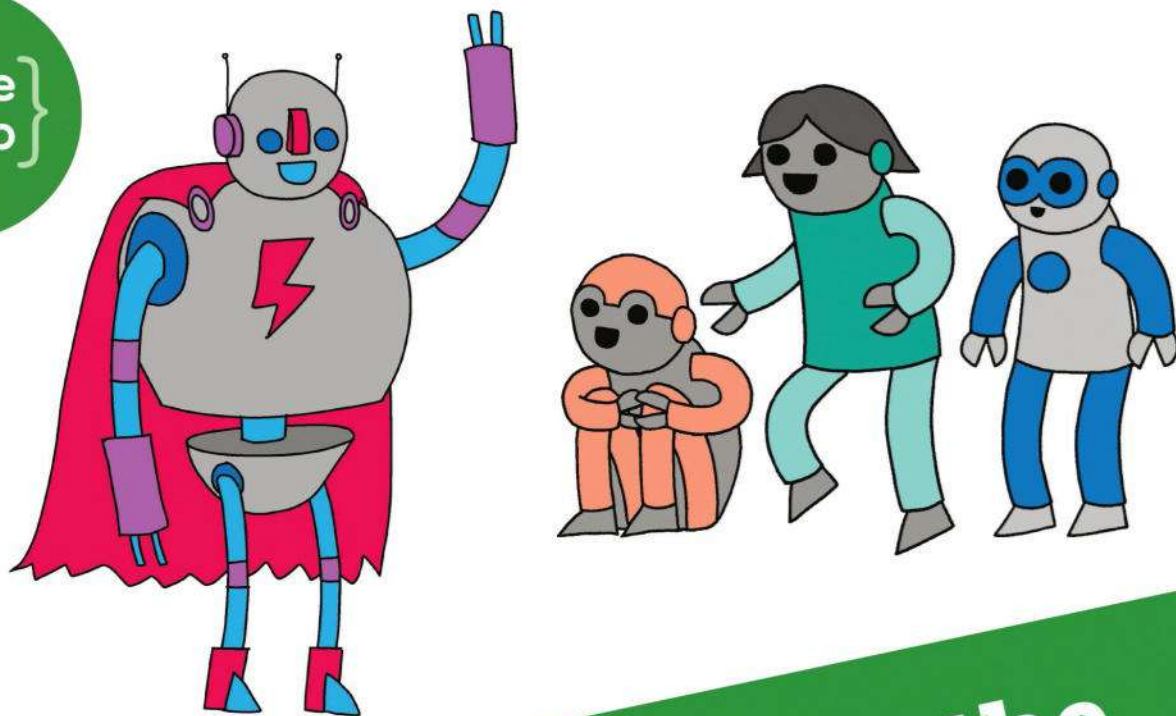
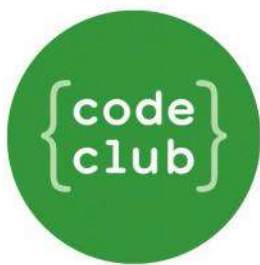


Head to Print Settings to set various default settings for your printer. By default, the printer prints in portrait orientation, but you can make it default to landscape. Similarly, the printer prints documents serially, but you can ask it to print in the reverse order with the last page first.

In the General options, you can select all kinds of media sizes. To change the appearance of your prints, expand the Printout Appearance options to increase or decrease the brightness of the prints.

If you print images, navigate to the Image Printing options, from where you can choose the position of the image on the page. By default, images are centred on the page, but if you know what you're doing, you can position the images to be anywhere from top-left to bottom-right. **LXF**

If your printer supports it, you can use the HP Print utility to print, scan and fax documents. It can also check the printer's ink or toner level, and display and control the printer queue.



**Can you help inspire the
next generation of coders?**



Code Club is a nationwide network of volunteer-led after school clubs for children aged 9-11.

We're always looking for people with coding skills to volunteer to run a club at their local primary school, library or community centre for an hour a week.

You can team up with colleagues, a teacher will be there to support you and we provide all the materials you'll need to help get children excited about digital making.

There are loads of ways to get involved!
So to find out more, join us at **www.codeclub.org.uk**

PYTHON

Code your own Wordle-like game

Keeping his grey matter well tested, **Matt Holder** creates a Wordle-style game and hopes the lawyers don't ask too many questions.



OUR
EXPERT

Matt Holder is an IT professional of 15 years, Linux user for over 20 years, home-automation fan and self-professed geek.

QUICK TIP

The complete source code can be downloaded from <https://github.com/mattmole/LXF-Wordle>.

During late 2021, a game called *Wordle* was released to the world, and it became very popular incredibly quickly. The appeal of the game is its simplicity, the relatively short amount of time it takes to play and the innovative way in which your daily score can be shared with friends.

In this article, we are going to create a clone of the famous game, using around 100 lines of Python code. Before we start coding, though, let's introduce the game a bit more thoroughly.

Wordle (now owned by the New York Times) is centrally hosted and there is only one game to play each day. This means that the game doesn't take too much time and enables friends to compare results by everybody having the same goal to work towards.

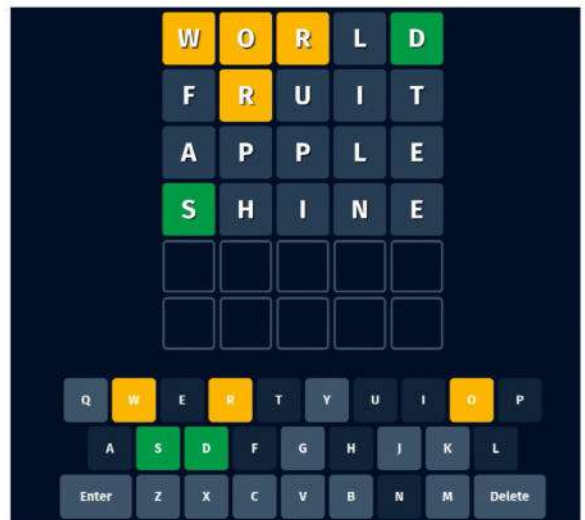
For each guess, the result is colour-coded in the following way: a green letter means that the letter in that position is correct; an orange letter means that the chosen letter is present in the word, but is currently in the wrong place; while a grey letter indicates that the letter isn't in the solution. Sharing your results with friends is clever because it doesn't give away the answer. A grid of coloured squares is generated, which can be shared. The colours represent the same as already described and shows the progress throughout the guesses.

Setting up for this project is easy. First of all, create a directory in which to store your code. Open your favourite IDE, install any Python add-ons that may be required, and create a new file, called **Wordle.py**. Make sure you have the rich library installed, which we will use to colour-code the output on the screen. You can install the library by opening a terminal and entering the following command:

```
$ pip install rich
```

The code will be structured by generating a class, which will contain the game logic, and then some code outside of the class, which allows the game to be interacted with. With that said, let's take a look at the first code sample:

```
from rich import console
from rich.prompt import Prompt
import random
```



Reactle is an open source clone of the Wordle game.

```
class Wordle:
    def __init__(self, wordFile="words_alpha.txt"):
        self.wordList = []
        self.loadFile(wordFile)
        self._randomWord = self.pickRandomWord()
        self.guessResults = None
        self.guessStatus = None
```

In the first three lines, we import two of the methods from the rich library and then the random library, which we will use later in the code. Next we declare a new class, which we call **Wordle**. We initialise the class using the `__init__` function, which is defined with two arguments. **Self** refers to the instance of the class being called (this will be done towards the end of the program) and **wordFile** is the name of the file that should be opened. If a copy of the class is created without an argument (**self** does not need to be used), the default of **words_alpha.txt** is used. The variable called **wordList** is used to store a list of words that are read in from the word file. Next we call the function that loads the file before calling the **pickRandomWord** function, which selects a random word from the **wordList** list. Finally, **guessResults** and **guessStatus**

are used to store the results of a guess. The format of **guessResults** will be discussed later.

As a brief aside, the **randomWord** variable is prefixed with an underscore character to denote that it should not be referenced directly. This leads us on nicely to getters and setters, which can be seen in the next code sample:

```
@property
def randomWord(self):
    return self._randomWord
```

```
@randomWord.setter
def randomWord(self, word):
    self._randomWord = word
```

In this code sample, we first of all declare a function called **randomWord**, which we can call a getter. This is a helper method that can be generated to return the value of a variable. While overkill in this scenario, imagine if you referenced a variable in 100 places and you decided that some processing needed to take place before accessing the value. Using this functionality, you can make the necessary changes in the function, instead of wherever the variable is referenced. The function uses the **@property** decorator to allow it to be considered as a property, rather than a function. We now define the same function name again, but this time with a decorator called **@randomWord.setter**. This allows this identically named function to be able to be used to set the value of **self._randomWord**. These functions are used as such: **a = a.randomWord** and **a.randomWord = 'wrist'**. Note how we don't need to use any brackets as we usually would when using functions.

```
def loadFile(self, wordFile):
    with open(wordFile, "r") as wordFile:
        for line in wordFile:
            word = line.strip()
            if len(word) == 5:
                self.wordList.append(word)
```

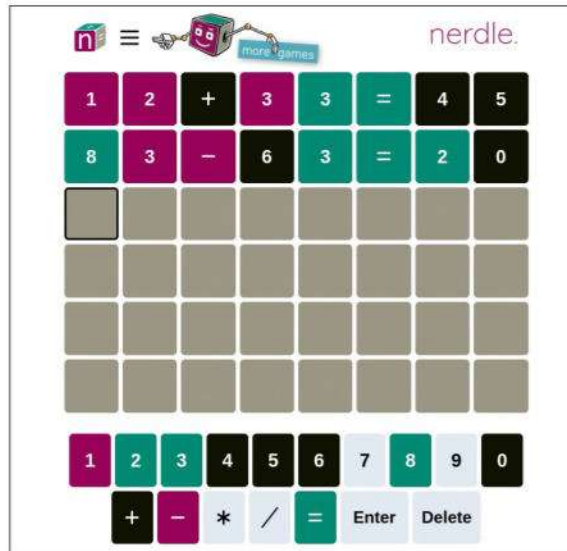
In the **loadFile** function, we first of all use **with** functionality to open our **wordFile** and allow us to process all lines within. The **for** loop is used to loop through each line and add relevant words to our **wordList** list. Within the **for** loop, we first of all strip any return characters from the start or end of the line and test if the word is five characters. Any five-character words are added to the list – other words are ignored.

```
def pickRandomWord(self):
    randomWord = random.choice(self.wordList)
    self.randomWord = randomWord
    return randomWord
```

The **pickRandomWord** function is very simple and returns a value, chosen at random from the **wordList**. For this we use the **choice** method, from the random library, which we imported at the beginning of the program. Also, note the line where we set the instance variable **self.randomWord** to the chosen word.

```
def checkWordGuess(self, guess):
    letterCounter = 0
    letterStatusCount = 0

    #Dictionaries can be used to store the status of the
    guessDict = {}
    usedLetterCount = {}
```



Nerdle is a similar game, which concentrates on simple arithmetic.

```
for letter in guess:
    #See how many times each letter appears in the
    guess
    letterCountInGuess = guess.count(letter)

    #See how many times each letter appears in the
    randomly chosen word
    letterCountInRandomWord = self.randomWord.
    count(letter)

    #Stick each letter in a dict to show when it has
    been used
    if letter not in usedLetterCount:
        usedLetterCount[letter] = 1
    else:
        usedLetterCount[letter] += 1

    #If the letter does not appear in the randomly
    chosen word, mark it as such
```

QUICK TIP

A couple of alternatives:
Nerdle can be played at <https://nerdlegame.com>
Qwordle can be played at <https://qwordle.bhat.ca>

» ALTERNATIVE GAMES

Once **Wordle** took the world by storm, plenty of alternatives also sprung up. The first, and our personal favourite, is **Nerdle**. This game takes the **Wordle** concept, but instead of displaying five spaces for letters, it instead shows eight spaces, which are filled in with numbers and arithmetic symbols. The aim of this version is to reproduce the chosen mathematical equation in six moves.

Another word game is **Qwordle**, which describes itself as the quantum version of **Wordle**. The aim is to guess two words in the same six guesses. Once a guess has been entered, the letters are coloured in a way that represents both the words being guessed.

As the name suggests, **Globe** takes the general concept of the game, but utilises the globe to show a country at random and the aim is to guess the country being shown. We are particularly terrible at this one.

Also changing themes somewhat, **Heardle** takes the concept of six guesses and provides snippets of a song to listen to, which becomes easier the more it is listened to.

Finally, **Waffle** is a game that takes the crossword puzzle concept and fuses it with **Wordle**, allowing a limited number of guesses to complete the small crossword.



QUICK TIP

An open source clone of *Wordle* can be played here: <https://modem7.github.io/react-wordle/>

```

        if letter not in self.randomWord:
            guessDict[letterCounter] = {"letter":letter, "reason": "letterNotInChosenWord", "string":f"[red]{letter}[/red]"}
        else:

            #Now we need to check how many times a
            letter is in the guess vs the chosen word
            #and act accordingly
            if usedLetterCount[letter] <=
            letterCountInGuess:
                if letter == self.randomWord[letterCounter]:
                    guessDict[letterCounter] = {"letter":letter,
                    "reason": "letterCorrectPosition", "string":f"[green]{letter}[/green]"}
                letterStatusCount += 1
            else:
                guessDict[letterCounter] = {"letter":letter,
                "reason": "letterWrongPosition", "string":f"[yellow]{letter}[/yellow]"}
                elif letterCountInGuess >=
                letterCountInRandomWord and
                usedLetterCount[letter] >=
                letterCountInRandomWord:
                    guessDict[letterCounter] = {"letter":letter,
                    "reason": "letterNotInChosenWord", "string":f"[red]{letter}[/red]"}

            letterCounter += 1
            self.guessResults = guessDict
            self.guessStatus = False
            if letterStatusCount == len(self.randomWord):
                self.guessStatus = True
            return guessDict, self.guessStatus
    
```

The **checkWordGuess** function is where the majority of the program's logic takes place and is where we need to check the guess against the chosen word and return feedback to the user. Later in this function, we return a colour-coded string to the main part of the program. It could be argued that this is out-of-scope of the **Wordle** class as its main aim is to perform the game logic, not provide the user interface. To begin with in this function, we define two variables, called **letterCount** and **letterStatusCount**, which we both set to **0**. The **letterCount** variable is used to determine how many letters through the guess we are and **letterStatusCount** is used to mark how many letters have been guessed correctly. If all five are correct, we can then mark the **guessStatus** variable as **true**, to indicate a completed game.

On the next two lines we create two variables, one called **guessDict** and the other called **usedLetterCount**. These are used to store the results of the guess and how many times each letter is used. This last point is important, because if the random word is 'world' and the guess contains two instances of the letter r, we need to be able to determine that one r should be marked as incorrect.

We now begin a **for** loop, where we iterate through each letter in the guess. The **letterCountInGuess** variable stores the number of times the letter we are currently processing appears in the guess. We then need to count how many times the letter we are currently processing appears in the **randomWord**, and this is stored in the **letterCountInRandomWord**

```

*****
Welcome to Wordle - have fun!
*****
1) Generate random word
2) Guess the word
3) Quit
: 2
Enter your guess: shoes
shoes
Enter your guess: apple
apple
Enter your guess: world
world
Congratulations - you did it!
    
```

Our Wordle clone has a command-line interface.

variable. On the next four lines, we either add the letter being processed to the **usedLetterCount** dictionary, with a key set to the letter and a value set to one, or we increment the value if it already exists in the dictionary. This is so that we can account for the situation we discussed earlier, where the guess contains the same letter more times than it appears in the random word. The dictionary was declared outside of the **for** loop, which means it will not get wiped each time the loop progresses.

We're now in a position where we can give the user some feedback of their guess. For each letter within the guess, we return a dictionary, which contains three things. First of all, the letter is quite important, and then we add a string, which relates to whether the letter is present, in the correct position, or not included at all. The third item we provide is a string, containing the letter, with the correct colour code.

The first item we return is the instance where the letter being processed is not in the **randomWord** at all. We construct a dictionary and then add it to the **guessDict** list. We next include an **else** statement, so the next few tests are all carried out in the context of the letter being included in the random word.

The first test is to see whether the letter being processed is in the correct position. We do this by testing against the letter in the same position in the **randomWord** variable (using the **letterCounter** as a reference). A dictionary is then constructed and added to **guessDict**. We also increment **letterStatusCount** by one for every letter that is correct. We now utilise an **else** statement, which means that the letter is included in the word but not in the correct position. Another dictionary is constructed and added to **guessDict** again.

The final test in this block of code is to check for the case where the letter in case has not been included in the **randomWord**. In this instance, we construct the dictionary to state that the letter is not included in the word. The final thing we do within this loop is to increment the **letterCounter** variable.

The final lines within the class assign the **guessDict** list to the **self.guessResults** variable. We then set the **self.guessStatus** variable to either **True** or **False**, to indicate whether the guess is correct or not, before returning these two variables.

We now need to create the user interface for our program, which we shall do using the **Console** functionality of the rich library:

```
if __name__ == "__main__":
    console = console.Console()
    a = Wordle()

    console.print("*****")
    console.print("Welcome to Wordle - have fun!")
    console.print("*****")

    carryOn = True
    while carryOn:
        console.print("1) Generate random word")
        console.print("2) Guess the word")
        console.print("3) Quit")
        response = Prompt.ask("")
        if response == "1":
            a.pickRandomWord()
        elif response == "2":
            guessCount = 5
            while guessCount > 0:
                guess = Prompt.ask("Enter your guess")
                if guess.lower() not in a.wordList:
                    console.print("Your guess is not a dictionary word. Try again")
                else:
                    guessResult, guessStatus = a.checkWordGuess(guess.lower())
                    guessString = ""
                    for letterGuess in guessResult:
                        guessString += guessResult[letterGuess]
                    ["string"]
                    console.print(guessString)
                    if guessStatus == False:
                        guessCount -= 1
                        if guessCount == 0:
                            console.print("Bad luck - fingers crossed for next time!")
                            console.print(f"The selected word was: [green]{a.randomWord}[/green]")
                        else:
                            console.print("Congratulations - you did it!")
                            guessCount = 0
                    elif response == "3" or response.lower() == "q":
                        carryOn = False
```

In this code sample, we first of all test if the program has been called directly or from another program. We then create an instance of the rich **console** class and a copy of our **Wordle** class. The **console.print()** command is then used to provide some welcome information to the player.

The **carryOn** variable is used to keep the player within the **while** loop, which we define next. This means that the game carries on until the quit option has been selected. Again, the **console.print()** functionality is used to show instructions to the player. These instructions are printed every time the game starts again. The **response** variable is used to store the input from the user. If number three is entered, the **carryOn** variable is set to **False** and the **while** loop exits, which

» WORDLE GUI

Using the *Wordle* library that we have written and the GUI Zero library, it is relatively simple to create a GUI, which can be used in place of the console program. This is the power of creating a library and the interface code separately. Our GUI code is separated into multiple parts, with the first being importation of libraries and setting of variables/constants. Next we define functions, called **finishGame**, **newGame**, **resetGame** and **checkGuess**. These control what happens when the game ends, when the game starts, when a game is restarted and when a word guess is checked to see if it is correct.

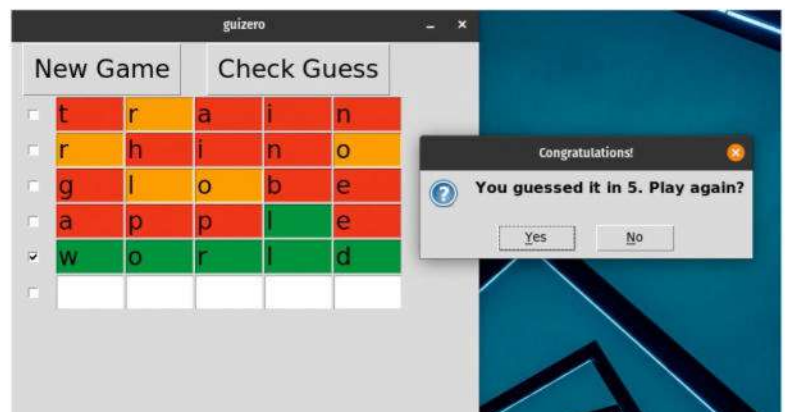
Next we define the necessary GUI elements to draw on the window (see screenshot, below). There is a grid of text boxes and these are all drawn on to the window by a pair of nested **for** loops. This allows a list of lists to be used as a 2D array and each cell can easily be referenced, to set the background colour or to read the contents. The concatenate values of a row are used to make up a guess and this is then fed to the **checkGuess** function. On creation, GUI elements are linked to functions, so that when a button is pressed, for example, a function is called. Finally, the window is made visible and the program can run. The code for the GUI program is included in the GitHub repository, which was linked to in the main body of the article.

also means that the program finishes. When selecting number one, the **pickRandomWord** function is called from the instance of our **Wordle** class.

If the user selects option two, they are asked to enter their guess. This is carried out within a **while** loop, which is set to test against a variable, initially set to five. For every guess that contains a dictionary word, this is decremented, then after five incorrect attempts, a message and the random word are printed. For each guess, the word is first tested against the word list to ensure it is a dictionary word. If not, a message is printed. Otherwise, the **checkWordGuess** function is called and the values returned are stored in **guessResult** and **guessStatus**. Each dictionary within the **guessResult** list represents a letter and the string information is added to a new string, which is then printed to screen. This contains a colour-coded version of the guess. If the **guessStatus** variable is set to **True**, a message congratulates the user.

We trust you have enjoyed this tutorial and it has given you some ideas for your own game clones. **LXF**

GUI Zero was used to create a GUI for our Wordle clone game.



» WE'RE CREATING THE LXF SHELL Subscribe now at <http://bit.ly/LinuxFormat>

How to build a 6502 assembler

David Bolton shows how to create a 6502 assembler in C#, highlighting the problems you need to overcome.



**OUR
EXPERT**

David Bolton worked as a games developer back in the '80s, programming in both Z80 and 6502 assembly language on ZX Spectrum, MSX, Amstrad CPC-64 and CBM-64/ CBM-16 home computers.

QUICK TIP

If you get a 'Symbol Lookup error' when you run *as65*, you may have to uninstall the Dotnet Snap and install Dotnet using *Apt*. This command removes it: `sudo snap remove --purge dotnet-sdk`. Then use the *Apt* command in the Quick Tip opposite.

This article shows you how to create a 6502 assembler, which we're calling *as65*. It won't be quite as powerful as the *xa65* assembler used in earlier articles (www.floodgap.com/retrotech/xa – see **LXF313**) but it'll be our own!

To get started, here are the steps it will go through when assembling:

1. Read source file and options. If there are any include files specified, read those in.
2. Do the first pass, evaluate labels.
3. If there were no errors in the first pass, do the second pass. Emit the machine code to disk.

We use two passes because of page-zero support. If we didn't have page zero, all addresses would be 16-bit. Accessing page-zero means instructions are one byte shorter and run slightly faster.

The problem is when a label is referred to before it is defined. If the memory location accessed is in page zero, the instruction will be two bytes long and this will affect the location of labels that follow.

The first pass builds up a dictionary of labels. By the time it has done the first pass, it knows the value of every label and can then generate correct code on the second pass.

Rules to follow

If you follow these rules, it will be less problematic.

1. Declare any zero-page memory locations before you use them. Think of these as the variables that your program will use.
2. Labels used for subroutines or branching forward can be called before they are declared. After pass one is done, the length of all instructions will have been determined.

You might ask why are we even using page zero, because instructions that access zero-page locations are two bytes long not three, and execute in three clock cycles not four.

LDA \$44 - \$A5 \$44

LDA \$4400 - \$AD \$00 \$44

Even with hardware sprites, writing fast code is very important. It's why game developers quickly switched to 6502 assembly language in the '80s, because Basic was just too slow.

Another technique that made the original assembler faster was having the source code file tokenised. All 56 opcodes plus a dozen compiler directives could be each assigned a single byte token. So, **LDA \$44**, which is seven bytes long as characters, could be converted into three bytes: a byte for **LDA**, a byte for **\$44** and an end-of-line character. Only labels were held in their original text form. The benefits of this were that when assembling, there was no need to parse opcodes like **LDA**; instead, a big jump table holding all tokens made it very quick to jump to the routine that handles the **LDA** opcode. *As65* doesn't use tokens but feel free to modify it and add them.

Addressing modes

The 6502 has the following eight addressing modes – the number is the value of the AddressMode enum for the addressing mode.

Addressing mode	Function
0 Zero-Page,X LDA \$40,X	Load A from address (\$40 + X)
1 Zero-Page LDA \$32	Load a from address \$32
2 Immediate LDA #\$05	Put 5 in A
3 Absolute LDA \$C000	Load A from address \$C000
4 Indirect,Y LDA (\$D0),Y	Load A from the address in ((\$D0) + Y)
5 Indirect,X LDA (\$D0,X)	Load A from the address in (\$D0+X)
6 Absolute,Y LDA \$D000,Y	Load A from address \$D000+Y

The two indirect addressing modes are slightly different. One works with the X register, while the other work with the Y register. Both only work with a page-zero address. However, the calculated 16-bit address is anywhere.

Indirect,X is called **Indexed Indirect** – for example, **LDA (\$10,X)**. The address is calculated by getting the



16-bit value from $\$10 + X$. So, if X is \$5 and the bytes at \$15 and \$16 are \$56 \$30, then this loads A from address \$3056. This only works in page-zero.

Indirect,Y is known as **Indirect indexed** – for example, **LDA (\$20),Y**. The address in \$20 is added to Y. So, if Y is \$6 and the address at \$20 is \$34 \$d0, this gets that address \$D034 and adds Y to it to give \$D04A, and the A register is loaded from that. This also only works in page-zero.

You use **indirect,X** with a table of 16-bit addresses, while **indirect,Y** enables you to index through memory pointed to by one address.

If x is 2, then the bytes at \$22/\$23 point to address \$0016. The bytes at \$24/\$25 hold \$0032, so if y is 2, the effective address is \$0034.

In our experience **indirect,Y** is more commonly used than **indirect,X**.

The **JMP** instruction can be done indirectly, a little like **Indirect,Y**, except there's no Y register involved. So, you can have either **JMP Address (absolute)** or **JMP (Address)**. It's the only instruction that does this, so it's not defined as an address mode.

Handling branches

There's one extra addressing mode that is relevant in these circumstances and we should mention, and that's for branches. All branch instructions are two bytes long and, regardless of the type of branch, the branch value is a single byte. If it's positive (1-127), then it branches forward, while if it's negative (128-255), it branches backwards.

The value it branches is relative to the instruction after the branch, so the code that follows is kind of pointless as it will always execute the next instruction at 4002 irrespective.

(\$20, x)		
\$20	\$21	0010
\$22	\$23	0016
\$24	\$25	0032
\$26	\$27	0076
(\$24), y		
\$32	06	
\$33	09	
\$34	1A	
\$35	65	

Indirection examples for X and Y.

If you've specified a value that is one byte long before the instruction is assembled, it will use the 8-bit value and be two bytes long (opcode then byte).

The assembled program is built up as a **List<Opcode>** as each line of the source is read.

By the end of the first pass, the program cycles through the list looking for any instruction that references a label. When it finds the label, it does a look up on the **Labels** dictionary and gets the value or reports an error.

All labels need to be evaluated before pass two so the code output will be the correct length. This is why you need to declare variables before you use them. If you don't, then the code referencing it will be initially assumed to be three bytes long but after the label is evaluated, it will turn out to be two bytes long and the assembler would need to move the instructions following up by one byte.

This is important, so if the assembler finds that an undeclared label is in zero-page, it will treat it as an error.

If the first pass doesn't generate any errors, then the second pass can generate the code. There

```
4000    BPL FRED
4002 FRED: CLC
```

Program architecture

The assembler starts by loading the source file into a **List<string>** then processes it line by line and handles code generation and calculation of branches in the two passes.

Labels are held in a **Dictionary<string,Label>**, where **Label** is a class holding the value, and a flag **Evaluated** to say whether the value has been evaluated yet. So, by default, all memory addressing instructions default to a 16-bit value and are three bytes long (opcode, then 16-bit address).

QUICK TIP

Being C#, you need to have .NET installed on your Linux box. On Ubuntu, this is done with the terminal command `sudo snap install dotnet-sdk-classic`, but if you prefer the Apt install, check <https://bit.ly/lxf319net>. For Ubuntu 24.04, use this command: `sudo apt-get update && sudo apt-get install -y dotnet-sdk-8.0`. Once installed, do this to check it and see which version it is – it'll likely be 8.0.100 or higher: `dotnet --version`.

» WHAT IS AN ASSEMBLER?

Compilers convert source code into machine code so that it can be executed. An assembler is similar, except the source code that's converted is assembly language. In our case, the assembly language is for 6502, the CPU in CBM-64, NES and other computers.

You'll find 6502 assembly language is probably the easiest to learn as the processor only has three registers (A, X and Y) and there are only 56 instructions. These are 1-3 bytes in length with the opcode; the first byte and the last two bytes form a 16-bit address running from 0-65535 (0x0000-0xFFFF in base 16). Many instructions can also access the first 256 bytes of memory, known as zero-page memory, and these are 2 bytes

long. The opcode is the first byte and the second is the address (0-255 or 0x00-0xFF). Some instructions such as **TXA** (transfer contents of X register to A register) are only one byte long.

As well as the opcodes, the assembler needs some way of allocating data as bytes (8-bit), words (16 bits) or strings (variable length ending with a zero byte). Also, the assembler has to be able to place code at any address in the 16-bit range and use symbolic names (labels) for memory addresses or values.

The compiler runs on Linux and takes a source file of 6502 assembly language and outputs a file of 8-bit machine code. Because it has to run on a CBM-64 (Vice emulator), the option to generate a PRG file is also included.

```
init_grid_outline:
    // (1) bottom
    lda #1 /// cell on
    ldx #0
!loop:
    sta $23a6,x // frozen + (23 * 40) +
    lda grid_outline_color
    sta $dba6,x // grey
    inx
    cpx #12
    bne !loop-
    // (2) left
    lda #$36 // 8192 + (1 * 40) + 14
    sta $f9
    lda #$20
    sta $fa
    lda #$36 // color ram
    sta $b0
    lda #$d8
    sta $b1
    jsr init_grid_outline_side
```

Example of 6502 code from tetris.asm.

» AN ANCIENT STORY...

Back in the 1980s, this author and a business partner started doing game conversion work for Ocean Software and US Gold in Manchester. They gave us a few Tatung Einsteins. This was a Z80 computer that ran CP/M and a source code copy of the Zen Z80 assembler and

editor. We were also doing 6502 development and I knew 6502 but not Z80. So, in order to learn Z80, I wrote a 6502 cross-assembler in Z80. Back then, there was no internet or libraries for advanced data structures like dictionaries. We used a simple mechanism to handle

labels – a 26x26 array of pointers. The first two letters were used to index (labels were upper-case alphabetic only) and then there was a linked list of label remnants. So, if the label was **ENTRY, E** and **N** were used to index the array, and **TRY** was stored in the list along with the value and a pointer to the next

label. A five-thousand-line game might have one or two hundred labels, and this mechanism worked well. Two weeks later, the assembler was finished and being used in anger. In the C# version discussed here, we're using a dictionary to hold instances of a **Label** class to hold labels.

are four areas that need to be looked at in more depth at this point:

1. **Parsing operands** – this means determining the addressing mode and calculating the length of the instruction.
2. **Expressions** – to keep things simple, the only expressions allowed are +/- a value.
3. **Compatibility with other assemblers** – this means using the same type of pseudo-opcodes for allocating memory (words, bytes and strings) with values.
4. **Generating a PRG file**

It's recommended that you don't put code in page zero because this may lead to code labels being marked as zero-page.

Parsing operands

Perhaps the most complex part of the assembler is evaluating the operand to determine the addressing mode and evaluating any expressions. The operand is the part after the opcode – the **#\$ff** in the **lda** below.

```
lda #$ff
```

This is fairly straightforward. Start by looking for a **#**, which means immediate mode. This example puts **\$ff** (255) into the A register.

If it doesn't exist, look for a value that starts with a **\$** or **0-9**. If it starts with a **\$**, then it's a hexadecimal number and is made up of **0-9** and **A-F** (upper or lower case), otherwise it's decimal but has to be in the range **0-65535** or it's an error.

```
sta $d019
```

This stores the A register in address **\$d019**.

Note: if it's hexadecimal and the length is four digits but the value is **\$00FF** or lower, then it's treated as a 16-bit value not an 8-bit value. If you want page zero addresses, use a two-digit hexadecimal number. This covers the case where you want to access memory in

either **Absolute,X** or **Absolute,Y** address mode that starts in page zero and carries on.

LDA \$00F4,X will add X to **\$00F4** and carry on past **\$0100**, while **LDA \$F4,X** will wrap around to **\$00** when it goes past **\$FF**. Note there is an **Absolute,Y** but no **Zero-Page,Y** addressing mode while there are both **Absolute,X** and **Zero-Page,X** modes.

If it is text, parse it as a label and look for an existing value in the **Labels** dictionary. If it doesn't exist, add it to the **Labels** dictionary.

Having obtained a value (or a label with an optional expression), a check is needed for **,Y** or **,X**.

If the first character is a left-bracket – **(** – then the mode is either indirect (JMP instructions only), indexed indirect (**value,X**) or indexed indexed (**value**),**Y**.

In this example, the a register is stored at the address in location **\$b0** and **\$b1** plus **y**.

```
sta ($b0),y
```

Labels are assigned a value in one of two ways:

1. The value is calculated from the initial address specified by ***= address** and then instructions up to the label.

```
LOOP:
```

```
Or
```

```
BLOCK1: LDA #$34
```

2. The label can be assigned a value with **=**, such as **FRED = 2**

In the first, it's important to correctly calculate the instruction lengths before **LOOP** as this affects the value of **LOOP**. Hence the need to enforce declaration of page zero location labels before they are accessed.

Expressions

To keep things simple, expressions are limited to just things such as **label +/- value**, where **value** is a number or another label. If you wish to modify it, this Stackoverflow question (<https://bit.ly/lxf319math>) has

links to various source libraries if you want to make the expression handling more complex.

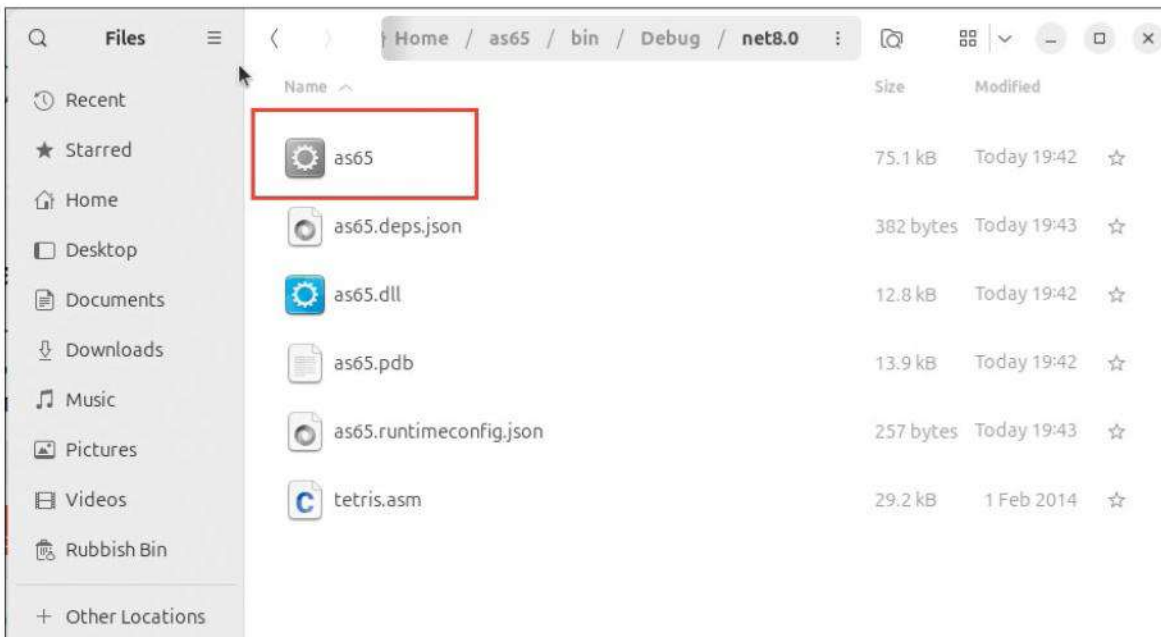
Compatibility

There is no standard for 6502 assembly language pseudo-opcodes, so we based this on the **xa65** assembler. Included in the **xa65** ZIP file is **xa.txt** that lists things like pseudo-opcodes. The ones used in **as65** are:

In an Ubuntu 24.04 terminal compiling as65 sources.

```
david@david-PC8:~/as65$ dotnet build as65.sln
Determining projects to restore...
Restored /home/david/as65/as65.csproj (in 50 ms).
as65 -> /home/david/as65/bin/Debug/net8.0/as65.dll

Build succeeded.
0 Warning(s)
0 Error(s)
```

List of files including the as65 executable.

Pseudo-opcode	Meaning
.byte	value1,value2,value3,...
.word	value1,value2,...
.asc	"text1","text2",...
.fill	length,fill-byte – creates table of specified length filled with fill-byte
*=	Set initial start address
#include "filename"	Where filename is the file to be included

It can be seen that xa65 is a lot more powerful, but the purpose of this tutorial is to show you how to create a 6502 assembler, not to recreate xa65.

Generating a PRG file

By default, the assembler just outputs a block of code and calculates all labels relative to the start address specified by a *= address statement.

However, if the **-p** flag is included, a PRG file is generated; this a block of code with a two-byte header in little endian format with the address that it's to be loaded at. The **-p** flag means the *= also writes the two-byte address at the start of the code in pass two.

Assembling

The assembler is called as65 and the C# source can be downloaded from GitHub (<https://bit.ly/lxf319zip>). To

compile the sources, unzip **as65.zip** into a folder, then open a terminal in that folder.

```
$ dotnet build as65.sln
```

To assemble a 6502-source file, move it into the **as65/bin/Debug/net8.0**, where you should see the compiled as65 executable. Then do this the command:

```
$ ./as65 filename
```

Unfortunately, this early version had a little bug and crashed... If you add **-p** on the end, it should generate a PRG file:

```
$ as65 filename -p
```

Another optional parameter **-v** will output verbose details. It provides extra output.

Opcodes

The **opcodes** class consists of an array of opcodes, each with the three-letter opcode and the value. For single-byte instructions such as **TAY**, **TAX**, **CLI** and so on, this is enough. The remaining one or two bytes are stored in the **Address** field.

This is left until pass two, once all labels have been evaluated. Error checking is done to ensure that branches are within 128 bytes in either direction.

To keep it simple, all the generated code is output in one block. A future change might be to allow multiple code blocks so that it can load into different places. But that would need a special loader.

As65 is a barebones first version that can be used to teach how to write assemblers as well as expanded upon to improve its performance and capabilities. **LXF**

QUICK TIP

A useful enhancement to as65 would be substituting the byte value of the opcode for the opcode text during the first pass, so it can be output on the second pass to speed up assembling. As65 incorporates optional aliases for some pseudo-opcodes such as "dsb" and .byte – both output byte data. This gives extra compatibility with other assemblers.

```
david@david-PC8:~/as65/bin/Debug/net8.0$ ./as65 tetris.asm
as65 6502 Assembler. Author D. Bolton
Unhandled exception. System.IndexOutOfRangeException: Index was outside the bounds of the array.
   at as65.Parameters..ctor(String[] args) in /home/david/as65/Parameters.cs:line 31
   at as65.Assembler.Run(String[] args) in /home/david/as65/Assembler.cs:line 23
   at as65.Program.Main(String[] args) in /home/david/as65/Program.cs:line 9
```

A less than 100% successful first run of the as65 assembler.

» **COME FINISH OUR ASSEMBLER!** Subscribe now at <http://bit.ly/LinuxFormat>

NEXT MONTH

LXF320
will be on sale
Tuesday 17th
September
2024

THE LINUX SMART HOME

We aim to keep the lights on the smarter way
with Home Assistant, Zigbee and low-cost IKEA kit.

Start using Linux

We test the best beginner distros to find out which can
get your friends and family up and running fastest!

Networking Linux

Get your head around networking basics, from Ethernet
ports to configuring your router and Wi-Fi connections.

Enigma made easy

Encrypt your own World War Two messages as we
explore how you can run your own Enigma machine.

Upgrade your graphics

Everything you need to know to get a faster, smarter,
better-installed graphics card in your Linux box.

Content of future issues subject to change. We might still be turning the lights on...

LINUX

The #1 open source mag

FORMAT

Future Publishing Limited, Quay House, The Ambury, Bath, BA11UA
Email contact@linuxformat.com

EDITORIAL

Editor-in-chief Neil Mohr

Art editor Fraser McDermott

Production editor Katharine Davies

Content director Marc Chacksfield

Group art director Warren Brown

Editorial contributors

Paul Alcorn, Mats Tage Axelsson, Mike Bedford,
Denise Bertacchi, Jonni Bidwell, David Bolton, Neil Bothwick,
Stuart Burns, Sam Dawson, Shane Downing, Nate Drake,
Tam Hanna, Matt Holder, Jon Masters, Dave Meikleham,
Nick Peers, Les Pounder, Michael Reed, Mayank Sharma,
Shashank Sharma, Jake Tucker, Sean Webster, Mike Williams

Cover illustration Magictorch.com

Ubuntu is a trademark of Canonical Limited.

We are not endorsed by or affiliated with Canonical Limited or the Ubuntu project.

Raspberry Pi is a trademark of the Raspberry Pi Foundation.

Tux credit: Larry Ewing (lewing@isc.tamu.edu).

Linux Mint is copyrighted 2006 and trademarked through the Linux Mark Institute.

Content production Adequate Media Limited

ADVERTISING

Commercial sales director Clare Dove
clare.dove@futurenet.com

Advertising director Lara Jaggon
lara.jaggon@futurenet.com

Account director Andrew Tilbury
andrew.tilbury@futurenet.com

INTERNATIONAL LICENSING

Head of print licensing Rachel Shaw

Linux Format is available for licensing and syndication.

To find our more contact us at licensing@futurenet.com

or view our content at www.futurecontenthub.com

NEW SUBSCRIPTIONS & BACK ISSUES

Web www.magazinesdirect.com

UK 0330 333 1113 World +44 (0) 330 333 1113

EXISTING SUBSCRIPTIONS

Web www.mymagazine.co.uk

UK 0330 333 4333 World +44 (0) 330 333 4333

Subscription delays: Please allow up to seven days before contacting us
about a late delivery to help@magazinesdirect.com

CIRCULATION

Newstrade & retail category director Ben Oakden

PRODUCTION AND DISTRIBUTION

Group head of production Mark Constance

Production manager Nola Cokely

Senior ad production manager Jo Crosby

Digital editions manager Jason Hudson

THE MANAGEMENT

Managing director technology group Paul Newman

Global head of design Rodney Dive

Commercial finance director Tania Brunning

Printed by William Gibbons & Sons

Distributed by Marketforce UK

121-141 Westbourne Terrace, London W2 6JR. www.marketforce.co.uk

For enquiries email: mfcommunications@futurenet.com

Order and access back issues: If you are an active subscriber, you have instant access to back issues through your iOS or Android devices. Your digital magazine entitlement is available at no additional cost and no further action is required. Pocketmags library may not have access to the full archive of digital back issues. You will only be able to access the digital back issues as long as you are an active subscriber.

To purchase single back issues (print format only): Visit www.magazinesdirect.com (click on Single Issues tab) or email help@magazinesdirect.com. For further help call +44 (0) 330 333 1113. Lines are open Mon-Fri 8.30am-7pm and Sat 10am-3pm UK time. Magazinesdirect.com is owned and operated by Future Publishing Limited.

BAR rates for Linux Format: £84.37 for UK, £171 for Europe, \$194 for USA, £149 for rest of world.

Linux® is the registered trademark of Linux Torvalds in the US and other countries. GNU/Linux is abbreviated to Linux throughout for brevity. Where applicable, code printed in this magazine is licensed under the GNU GPL v2 or later. See www.gnu.org/copyleft.html. All copyrights and trademarks are recognised and respected.

Disclaimer: All contents © 2024 Future Publishing Limited or published under licence. All rights reserved. No part of this magazine may be used, stored, transmitted or reproduced in any way without the prior written permission of the publisher.

Future Publishing Limited (company number 2008885) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA11UA. All information contained in this publication is for information only and is, as far as we are aware, correct at the time of going to press. Future Publishing Limited cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/services referred to in this publication. Apps and websites mentioned in this publication are not under our control. We are not responsible for their contents or any other changes or updates to them. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to us, you warrant that you own the material and/or have the necessary rights/permissions to supply the material and you automatically grant Future Publishing Limited and its licensees a licence to publish your submission in whole or in part in any/all issues and/or editions of publications, in any format published worldwide and on associated websites, social media channels and associated products. Any material you submit is sent at your own risk and, although every care is taken, neither Future nor its employees, agents, subcontractors or licensees shall be liable for loss or damage. We assume all unsolicited material is for publication unless otherwise stated, and reserve the right to edit, amend and adapt all submissions. All contents in this magazine are used at your own risk. We accept no liability for any loss of data or damage to your systems, peripherals or software through the use of any guide.

We are committed to only using magazine paper derived from responsibly managed, certified forestry and chlorine-free manufacture. The paper in this magazine was sourced and produced from sustainable managed forests, conforming to strict environmental and socioeconomic standards.

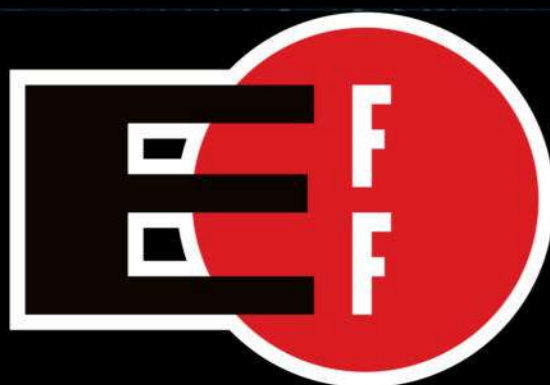
Future is an award-winning international media group and leading digital business. We reach more than 57 million international consumers a month and create world-class content and advertising solutions for passionate consumers online, on tablet & smartphone and in print.



Future plc is a public
company quoted on the
London Stock Exchange
(symbol: FUTR)
www.futureplc.com

Chief Executive Officer Jon Steinberg
Non-Executive Chairman Richard Huntington
Chief Financial and Strategy Officer Penny Larkin-Brand
Tel +44 (0) 1225 442 244





The Electronic Frontier Foundation is the leading nonprofit organization defending civil liberties in the digital world. Founded in 1990, EFF champions user privacy, free expression, and innovation through impact litigation, policy analysis, grassroots activism, and technology development. We work to ensure that rights and freedoms are enhanced and protected as our use of technology grows.

EFF.ORG

ELECTRONIC FRONTIER FOUNDATION

Protecting Rights and Promoting Freedom on the Electronic Frontier



**THE
BRAIN
TUMOUR
CHARITY**

A CURE CAN'T WAIT

**BRAIN TUMOURS
MOVE FAST.
WITH YOUR HELP,
WE CAN TOO!**

We're working to create a future where brain tumours are curable.
We urgently need your help to accelerate research.

Text DEFEAT5 to 70507 to donate £5, please help us to find a cure.

thebraintumourcharity.org

© The Brain Tumour Charity 2020. Registered Charity in England and Wales
(1150054) and Scotland (SC045081)



Registered with
**FUNDRAISING
REGULATOR**